



**Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Curso de Engenharia Eletrônica**

**DISPOSITIVO DE AUXÍLIO À LOCOMOÇÃO DE
DEFICIENTES VISUAIS UTILIZANDO FUSÃO
SENSORIAL EM ARQUITETURAS
RECONFIGURÁVEIS**

**Autor: Lays Alves Fernandes de Faria
Orientador: Prof. Dr. Daniel M. Muñoz Arboleda**

**Brasília, DF
2015**



Lays Alves Fernandes de Faria

**DISPOSITIVO DE AUXÍLIO À LOCOMOÇÃO DE DEFICIENTES VISUAIS
UTILIZANDO FUSÃO SENSORIAL EM ARQUITETURAS RECONFIGURÁVEIS**

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Orientador: Prof. Dr. Daniel Mauricio Muñoz Arboleda

**Brasília, DF
2015**

Faria, Lays Alves Fernandes de.

Dispositivo de Auxílio à Locomoção de Deficientes
Visuais Utilizando Fusão Sensorial em Arquiteturas
Reconfiguráveis / Lays Alves Fernandes de Faria.
Brasília: UnB, 2015. 70 p. : il. ; 29,5 cm.

Monografia (Graduação) – Universidade de Brasília
Faculdade do Gama, Brasília, 2014. Orientação: Prof. Dr.
Daniel Mauricio Muñoz Arboleda.

1. Arquiteturas Reconfiguráveis. 2. Filtro de Kalman. 3.
Deficientes Visuais I. Mauricio Muñoz Arboleda, Daniel. II. Dr.

CDU Classificação



Dispositivo de Auxílio à Locomoção de Deficientes Visuais utilizando Fusão Sensorial em Arquiteturas Reconfiguráveis

Lays Alves Fernandes de Faria

Monografia submetida como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica da Faculdade UnB Gama - FGA, da Universidade de Brasília, em (data da aprovação 27/11/15) apresentada e aprovada pela banca examinadora abaixo assinada:

Prof. Dr.: Daniel Mauricio Muñoz Arboleda, UnB/ FGA
Orientador

Prof. Dr. Gilmar Silva Beserra, UnB/ FGA
Membro Convidado

Prof. M.Sc. Gerardo Idrobo Pizo, UnB/ FGA
Membro Convidado

Brasília, DF
2015

AGRADECIMENTOS

Aos meus pais, Joaquim e Kátia, e ao meu irmão Cayo, pelo apoio incondicional em todos os momentos da minha vida.

Aos meus avós, tios e primos, por todo o carinho, apoio e incentivo ao longo dessa jornada.

Aos professores que fizeram parte da minha formação acadêmica, em especial ao Prof. Dr. Daniel Muñoz, pela colaboração e dedicação em suas orientações prestadas durante o desenvolvimento deste projeto.

A todos os meus amigos, pelos bons momentos de convivência e aprendizado durante todos esses anos de estudo.

RESUMO

De acordo com a Organização Mundial de Saúde (OMS), em dados publicados em outubro de 2013, existem cerca de 39 milhões de deficientes visuais no mundo. Devido a essa limitação, locomover-se em ambientes desconhecidos é uma grande dificuldade para esses indivíduos. Existem diversas pesquisas realizadas com o propósito de criar soluções e tecnologias que visam assistir à mobilidade de deficientes visuais. Atualmente, o professor Dr. Daniel Muñoz coordena um projeto de pesquisa que visa a construção de arquiteturas de *hardware* para um dispositivo de auxílio à locomoção em ambientes desconhecidos. Uma das propostas para este dispositivo possui formato semicircular para ser utilizado pelo usuário em sua cabeça, e é constituído de sensores de distância como o ultrassom e infravermelho e eletrônica embarcada baseada em dispositivo reconfigurável *Field Programmable Field Array* (FPGAs), visando a implementação de algoritmos para tratamento de dados, através da utilização do filtro de Kalman (FK), e também para direcionar o usuário a se desviar de obstáculos. Diante disto, este trabalho propõe a modelagem e descrição de arquiteturas de *hardware* destes algoritmos em linguagem VHDL. São apresentados, também, a descrição do problema a ser resolvido, a metodologia, a fundamentação teórica dos algoritmos escolhidos e modelagem do sistema, assim como elementos usados para os processos de simulação e validação. São expostos ainda os resultados de simulação em *software* e *hardware* dos modelos matemáticos dos sensores e algoritmos utilizados e a arquitetura de *hardware* proposta. Por fim, são descritas as conclusões obtidas, bem como sugestões para trabalhos futuros.

Palavras-chave: Arquitetura Reconfigurável. Deficientes visuais. Filtro de Kalman. Fusão Sensorial. Sensor. FPGA.

ABSTRACT

According to the World Health Organization (WHO), in data published at October 2013, there are approximately 39 million blind people in the world. Due this limitation, move inside an unknown place is a major difficulty for the visually impaired people. Several researches were made in order to create new technologies and solutions to help the mobility of these individuals. Currently, Dr. Daniel Muñoz coordinates a project that aims to build hardware architectures for a device that helps locomotion at unknown environments. One of the proposals is a device that has a semicircular shape, to be used at the head, consisting of distance sensors such as ultrasound and infrared, and onboard electronics based on Field Programmable Gate-Array (FPGA), which will implements data treatment algorithms, provided by the Kalman filter (FK), and to direct the user in order to avoid obstacles. This work proposes the modelling and description of hardware architectures of those algorithms in VHDL language. It also presents the problem description to be resolved, the methodology, the theoretical background of the chosen algorithms and the system modeling, as well as the elements used for simulating and validation processes. In addition, the partial results of software and hardware simulations of the mathematical models of the sensors and algorithms used and the proposed hardware architecture are presented. Finally, the conclusions are described as also suggestions for future work.

Keywords: Reconfigurable architecture. Visually impaired. Kalman filter. Sensor fusion. Sensor. FPGA.

LISTA DE FIGURAS

Figura 1. Solução para auxílio à locomoção em formato semicircular.	14
Figura 2. Tecnologias assistivas à locomoção de deficientes. (a) bengala eletrônica. (b) luva eletrônica (c) robô cão-guia.	20
Figura 3. Fluxo de algoritmo do algoritmo do Filtro de Kalman (Kim, 2011).	22
Figura 4. FPGA Nexys 4 (Digilent).	29
Figura 5. Fluxo de projeto de FPGAs, adaptado de (Kilts, 2007).	30
Figura 6. Considerações para simulações: disposição das paredes.	32
Figura 7. Sensor ultrassom (Maxsonar LV-EZ1-MB1010, 2005).	37
Figura 8. Curva de resposta do sensor infravermelho em função da distância (GP2Y0A02YK, 2001).	37
Figura 9. Sensor infravermelho (Robotics-electronics).	38
Figura 10. Disposição dos sensores para a calibração (acima se encontra o sensor ultrassom e abaixo o infravermelho).	39
Figura 11. Calibração dos sensores.	39
Figura 12. Curva de calibração do sensor ultrassom e modelo matemático.	40
Figura 13. Curva de calibração do sensor infravermelho e modelo matemático.	40
Figura 14. Resultado da implementação do FK para o sensor ultrassom.	43
Figura 15. Resultado da implementação do FK para o sensor infravermelho.	43
Figura 16. Resultado da fusão sensorial.	44
Figura 17. Algoritmo implementado em MATLAB para verificação do funcionamento do algoritmo de reação.	45
Figura 18. Arquitetura para fusão sensorial em diagrama de blocos.	46
Figura 19. Arquitetura do modelo do sensor ultrassom.	46
Figura 20. Arquitetura do modelo do sensor infravermelho.	47
Figura 21. Arquitetura do algoritmo do filtro de Kalman.	47
Figura 22. Arquitetura do algoritmo de fusão sensorial.	48
Figura 23. Diagrama de estados do algoritmo de reação.	49
Figura 24. Diagrama de estados do comparador.	50
Figura 25. Arquitetura Final em diagrama de blocos.	52
Figura 26. Erro quadrático médio entre resultados de <i>hardware</i> e <i>software</i> para os modelos do sensor ultrassom. MSE = 2,53E-8.	57
Figura 27. Erro quadrático médio entre resultados de <i>hardware</i> e <i>software</i> para o modelo do sensor infravermelho. MSE = 3,39E-7.	57
Figura 28. Erro quadrático médio entre resultados de <i>hardware</i> e <i>software</i> para o modelo do FK para o sensor ultrassom. MSE = 6,97E-4.	58

Figura 29. Erro quadrático médio entre resultados de <i>hardware</i> e <i>software</i> para o modelo do FK para o sensor infravermelho. MSE = 1,71E-2.	58
Figura 30. Erro quadrático médio entre resultados de <i>hardware</i> e <i>software</i> para o modelo de fusão sensorial. MSE = 4.12E-8.	59
Figura 31. Erro quadrático médio entre resultados de <i>hardware</i> e <i>software</i> para a arquitetura geral de fusão sensorial. MSE = 1,02.....	60
Figura 32. Erro quadrático médio entre resultados de <i>hardware</i> e <i>software</i> para a arquitetura do algoritmo de reação. MSE = 1.29E-7	61

LISTA DE TABELAS

Tabela 1. – Parâmetros do algoritmo do filtro de Kalman utilizados em simulações.	26
Tabela 2. – Parâmetros do algoritmo de fusão sensorial utilizados nas simulações.	28
Tabela 3. – Acionamentos dos atuadores quando um sensor detecta um obstáculo: D representa desligado e L representa ligado.	34
Tabela 4. – Tempo de execução dos passos do algoritmo de fusão sensorial.	41
Tabela 5. – Tempo de execução dos passos do algoritmo do Filtro de Kalman.	42
Tabela 6. – Codificação para cada sensor: U representa o sensor ultrassom e I o sensor infravermelho.	53
Tabela 7. – Resultados de síntese e performance das arquiteturas de <i>hardware</i> .	55
Tabela 8. – Erro quadrático médio das arquiteturas propostas.	56
Tabela 9. – Tempo de execução das arquiteturas de <i>hardware</i> .	62

LISTA DE SIGLAS

ADC	Conversor analógico-digital
ASICs	Circuitos integrados de aplicações específicas
CLBs	Blocos lógicos configuráveis
DCMs	<i>Data Clock Management</i>
DSPs	<i>Digital Signal Processing</i>
FFs	Registradores <i>Flip-Flops</i>
FK	Filtro de Kalman
FPGAs	<i>Field Programmable Gate Arrays</i>
GPS	<i>Global Positioning System</i>
HDLs	<i>Hardware Description Languages</i>
IP cores	Propriedade intelectual
LUTs	<i>Look-Up Tables</i>
OMS	Organização Mundial da Saúde
PLL	<i>Phased Locked Loop</i>
PSD	<i>Position Sensing Detector</i>
PWM	<i>Pulse-Width Modulation</i>
RAM	<i>Random Access Memory</i>
VHDL	<i>VHSIC Hardware Description Language</i>
VHSIC	<i>Very-High-Speed Integrated Circuit</i>

SUMÁRIO

1 INTRODUÇÃO	12
1.1 DESCRIÇÃO DO PROBLEMA	13
1.2 JUSTIFICATIVA.....	15
1.3 OBJETIVOS.....	16
1.3.1 Objetivo geral.....	16
1.3.2 Objetivos específicos	16
1.4 METODOLOGIA	17
1.5 ORGANIZAÇÃO DO TRABALHO.....	18
2 MARCO TEÓRICO	19
2.1 TECNOLOGIAS ASSISTIVAS VOLTADAS À LOCOMOÇÃO DE DEFICIENTES VISUAIS	19
2.2 FILTRO DE KALMAN	20
2.2.1 Introdução.....	20
2.2.2 Filtro de Kalman simples para filtragem de ruído.....	22
2.2.3 Fusão sensorial.....	26
2.3 ARQUITETURAS RECONFIGURÁVEIS - FPGA	28
2.3.1 Fluxo de projeto em FPGAs.....	29
3 IMPLEMENTAÇÕES	31
3.1 ALGORITMO DE REAÇÃO	31
3.2 ASPECTOS GERAIS.....	34
3.2.1 Ambientes e programas para implementações de <i>software</i>	35
3.2.2 Ambiente e programa para implementações de Hardware	35
3.2.3 Sensor ultrassom	36
3.2.4 Sensor infravermelho PSD	37
3.2.5 Kit FPGA.....	38
3.3 CALIBRAÇÃO DOS SENSORES	38
3.4 IMPLEMENTAÇÕES EM HARDWARE E SOFTWARE.....	41
3.4.1 Implementações em <i>software</i>	41
3.4.2 Implementações em <i>hardware</i>	45
4 METODOLOGIA E IMPLEMENTAÇÕES PARA TESTES NA FPGA	53
4.1 IMPLEMENTAÇÕES PARA TESTES NA FPGA	53
4.2 METODOLOGIA DE TESTES	54
5 RESULTADOS.....	55
5.1 RESULTADOS DE SÍNTESE PARA IMPLEMENTAÇÕES EM VHDL	55
5.2 COMPARAÇÃO ENTRE ARQUITETURAS DE <i>HARDWARE</i> E <i>SOFTWARE</i>	55
5.2.1 Modelos do sensor ultrassom	57
5.2.2 Modelos do sensor infravermelho	57
5.2.3 Modelos do algoritmo de FK	58
5.2.4 Modelos do algoritmo de fusão sensorial.....	59
5.2.5 Arquitetura geral da fusão sensorial	60
5.2.5 Arquitetura do algoritmo de reação.....	61
5.3 TEMPO DE EXECUÇÃO	61
6 CONCLUSÕES	63

REFERÊNCIAS.....	65
ANEXO	68

1 INTRODUÇÃO

Dados recentes da Organização Mundial da Saúde (OMS), atualizado em outubro de 2013, apontam que há cerca de 39 milhões de deficientes visuais no mundo (Organização Mundial da Saúde, 2013), o que gera diversas consequências para estas pessoas, entre elas a dificuldade de locomoção, o que os tornam dependentes de métodos que ajudem na sua mobilidade sem o auxílio de outros indivíduos. Entre tais métodos pode-se mencionar o uso de cão-guia e de bengalas (Hoffmann; Seewald, 2003).

O cão-guia demonstra ser um excelente elemento de assistência, pelo treinamento que recebe, tornando-o capaz de prever acidentes e encontrar obstáculos que uma bengala comum não detectaria, geralmente com objetos localizados fora do alcance de meio metro a frente do indivíduo e acima da cintura. Entretanto, é um privilégio para poucos, já que o treinamento é caro e longo, com baixo índice de sucesso, além da pouca disponibilidade desses animais para suprir a demanda dos necessitados (Sullivan, 2013; Santos et al., 2010). Por outro lado, o cão-guia oferece assistência para evitar choque contra obstáculos, porém não é capaz de reconhecer ou guiar o deficiente a um local desejado em ambientes desconhecidos (Muñoz et al., 2010).

Embora existam as desvantagens anteriormente citadas, as bengalas são os instrumentos mais econômicos e eficientes utilizados pelos deficientes visuais. Tais dispositivos oferecem ao usuário a sensação de caminhar passando a mão no solo, funcionando também como proteção a possíveis choques, já que auxiliam no equilíbrio da pessoa (Santos et al., 2010).

É evidente que, apesar da existência de tais mecanismos de auxílio, a locomoção de deficientes visuais em ambientes desconhecidos é um fator que gera desconforto, não apenas do ponto de vista físico, como também problemas de inclusão social, qualidade de vida e sensação de liberdade de cada indivíduo (Hersh; Johnson, 2008).

1.1 DESCRIÇÃO DO PROBLEMA

Tendo em vista as inconveniências dos instrumentos de auxílio apresentadas, é possível inferir que existe o risco constante de choques contra obstáculos por parte dos deficientes visuais, além do problema evidente de locomoção em ambientes desconhecidos. Esses métodos demonstram severas limitações, já que se faz necessário o treinamento do usuário para o uso adequado.

Com os avanços científicos e processamento cada vez mais rápidos dos computadores, surgem, também, novas tecnologias com o objetivo de ajudar na mobilidade de deficientes, tais como bengalas eletrônicas e robôs cão-guia eletrônicos, fazendo uso de recursos como câmeras, *Global Positioning System* (GPS), reconhecimento ótico e redes de computadores, sinais sonoros, sistemas de substituição de visão, nanotecnologia, sensores de distância, entre outros (Singhal; Kaur; Yadav, 2013).

Entretanto, mesmo com o avanço da computação, algoritmos envolvidos no aspecto da mobilidade, com a finalidade de detecção de obstáculos, baseadas na utilização de sensores de distância tais como o infravermelho e o ultrassom, requerem complexos processamentos computacionais por lidar com processamento sensorial de imagem e posição no ambiente, sendo necessário, portanto, usar plataformas de sistemas embarcados de alto desempenho (Gharpure; Nicholson; Pavithran, 2004; Roentgen et al., 2008).

Considerando o problema descrito, o professor Daniel Muñoz coordena atualmente um projeto de pesquisa que visa o desenvolvimento de arquiteturas de *hardware* dedicadas para a elaboração de um dispositivo de auxílio à locomoção de deficientes visuais.

Uma das propostas de solução dentro desse projeto consiste na construção de um dispositivo em formato semicircular para auxílio ao movimento. É constituído de 7 pares de sensores infravermelho e ultrassom; um sistema embarcado para tratamento de dados provenientes dos sensores e processamento de algoritmos para detecção de obstáculos; e atuadores piezoelétricos para informar ao usuário a direção a ser tomada por meio de vibração, que será mais acentuada conforme o usuário se

aproxime de um obstáculo. A disposição dos componentes do projeto é dada conforme mostra a Fig. (1).

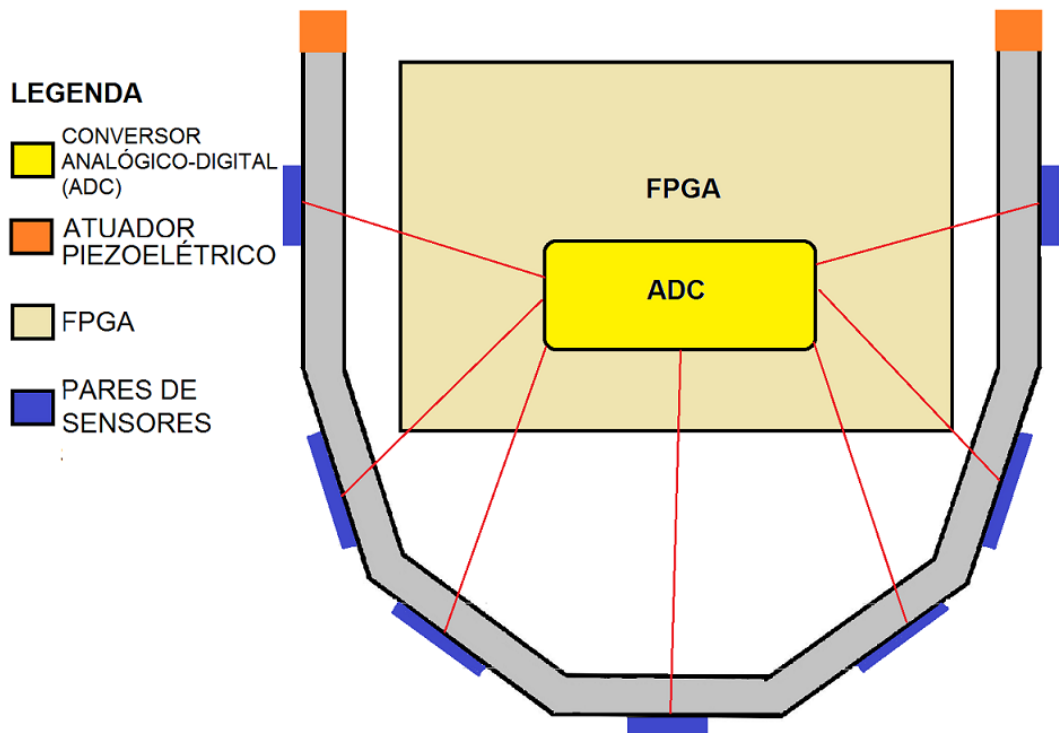


Figura 1. Solução para auxílio à locomoção em formato semicircular.

O sistema embarcado é dividido em três áreas de funcionamento:

- a) A primeira deve ser capaz de filtrar ruídos provenientes dos sensores, cuja proposta de solução baseia-se no uso do Filtro de Kalman (FK), um algoritmo bastante utilizado para filtragem de ruídos, e, assim, melhorar a informação dada pelos sensores.
- b) A segunda parte consiste em unir os dados dos dois sensores, já filtrados, através da fusão sensorial, utilizando para isso um algoritmo também baseado no FK.
- c) A terceira parte consiste em processar os dados tratados na segunda parte por meio de algoritmo de reação, a fim de produzir uma saída tal que acione os atuadores piezoelétricos para indicação da direção a ser tomada pelo deficiente.

1.2 JUSTIFICATIVA

Diante de todo o contexto abordado, surge a necessidade de pesquisas em torno de tecnologias assistivas ao problema discutido, onde a engenharia eletrônica se apresenta como uma área com muitos recursos e conhecimentos úteis para a criação de soluções com a finalidade de auxiliar e melhorar a locomoção autônoma desse grupo de pessoas. Entretanto, poucos são os estudos realizados nesse âmbito até o presente momento, o que pode ser justificado pela complexidade de algoritmos que levam em conta diversos fatores aleatórios que interferem diretamente na medição, tratamento e estimação da informação sensorial (Sass; Schmidt, 2010; Meyer-Baese, 2004).

As informações sensoriais possuem ruídos intrínsecos, sendo impossível prever a quantidade ou tipo de ruído associado a uma medida em um instante de tempo. Nos sistemas em tempo real, faz-se necessária a obtenção de várias medidas que, combinadas, são utilizadas para estimar o estado do sistema, e para isso, o tratamento para eliminação dos ruídos demonstra-se de vital importância para a estimação da medida correta do estado do sistema.

Portanto, para o tratamento de dados lidos dos sensores de distância, bem como a fusão sensorial deles, destaca-se o uso do Filtro de Kalman (FK). Este algoritmo foi criado por Rudolph Emil Kalman, apresentado em artigo publicado no ano de 1960, no qual ele descreve o filtro como “um processo recursivo para solucionar problemas lineares relacionados à filtragem de dados discretos” (Aiube, 2005).

O FK é constituído por um conjunto de equações que estimam uma medida diminuindo o seu erro quadrático ao longo das interações. Basicamente, o FK estima o estado instantâneo de um sistema dinâmico linear usando medidas ruidosas relacionadas ao comportamento de tal sistema. Ele possui como objetivo usar grandezas ruidosas para gerar resultados mais próximos aos valores reais que as grandezas deveriam apresentar. É uma das ferramentas mais conhecidas e utilizadas para estimação de medidas estocásticas de sensores ruidosos e fusão sensorial (Negenborn, 2003).

É importante ressaltar que a escolha dos modelos matemáticos para o tratamento de dados provenientes dos sensores e a implementação eficiente dos

algoritmos envolvidos influenciam diretamente nas características físicas e técnicas do dispositivo, tais como tamanho, peso, desempenho e consumo de energia. Assim, o desenvolvimento de arquiteturas de *hardware* paralelas voltadas para a solução desse problema possui a vantagem de acelerar a execução dos algoritmos, tornando-os mais eficientes se comparado com implementações em software.

Dessa forma, o uso de dispositivos *Field Programmable Gate Arrays* (FPGAs) torna-se uma solução apropriada como sistema embarcado do dispositivo em forma semelhante a um par de óculos para a implementação em arquiteturas paralelas do Filtro de Kalman para filtragem e fusão de sensores. Dispositivos FPGAs permitem explorar o paralelismo intrínseco dos algoritmos, constituindo também plataformas de prototipagem de circuitos digitais, sendo possível realizar as fases de teste, verificação e validação do *hardware* (Sass; Schmidt, 2010).

1.3 OBJETIVOS

1.3.1 Objetivo geral

O presente trabalho visa implementar e validar uma arquitetura de *hardware* do Filtro de Kalman para estimar a distância entre os obstáculos e o usuário do sistema (deficiente visual), permitindo indicar a direção de movimento no intuito de evitar a colisão com os obstáculos no ambiente através de um algoritmo para a indicação de direção de movimento.

1.3.2 Objetivos específicos

Os objetivos específicos do trabalho estão relacionados a seguir:

- a) Estudo e escolha de um algoritmo para o Filtro de Kalman para filtragem de ruído e fusão sensorial de um par de sensores de medição de distância.
- b) Calibração e obtenção dos modelos matemáticos dos sensores de ultrassom e infravermelho.
- c) Implementação em MATLAB do Filtro de Kalman para validação da modelagem adotada.

- d) Implementação, caracterização e validação em FPGAs de uma arquitetura paralela do FK.
- e) Proposta de algoritmo de reação para indicar a direção a ser tomada pelo deficiente visual.
- f) Implementação, caracterização e validação em FPGAs de uma arquitetura paralela para o algoritmo de reação.

1.4 METODOLOGIA

No desenvolvimento deste trabalho foram utilizadas as metodologias *top-down* e *bottom-up* para fins de organização das fases do projeto. No início, utilizou-se a metodologia *top-down* para uma análise do problema e modelagem da arquitetura *hardware* em FPGA. No desenvolvimento da arquitetura *hardware*, utilizou-se a metodologia *bottom-up*, que consiste no desenvolvimento, teste e validação da arquitetura final e seus módulos menores que compõem o sistema.

As fases de desenvolvimento são as seguintes:

- a) Na fase 1 realizou-se o estudo do estado da arte do problema apresentado, do algoritmo FK para filtragem e fusão de sensores, bem como a modelagem dos parâmetros do algoritmo no contexto do projeto.
- b) Na fase 2 foi realizado o processo de calibração dos sensores, com auxílio do conversor analógico-digital (ADC) do microcontrolador Arduino com resolução de 10 bits para leitura de distâncias na faixa de 20 a 130 cm. Com os dados provenientes do conversor, foi usado o MATLAB para obter o modelo matemático de observação de cada sensor.
- c) Na fase 3 foram feitas as implementações em *software* dos modelos do FK para filtragem de ruído e fusão sensorial. Adicionalmente, foi feita uma implementação em linguagem C para análise de perfil, assim como a codificação no MATLAB para mostrar a implementação dos algoritmos em forma gráfica. O MATLAB também foi usado para criar os vetores de teste aleatórios no processo de verificação dos circuitos propostos.

- d) Na fase 4 realizou-se as descrições em VHDL dos componentes que compõem a arquitetura de fusão sensorial de *hardware* proposta, bem como a simulação e caracterização do circuito em termos de recursos de *hardware*, frequência de operação, tempo de execução e consumo de potência.
- e) Na fase 5 foi feita a validação da arquitetura proposta para a fusão sensorial, comparando os resultados obtidos em *software* e *hardware*.
- f) Na fase 6 foi realizada a proposta de algoritmo de reação e modelagem via MATLAB para validação do funcionamento do algoritmo.
- g) Na fase 7, de maneira semelhante a fase 4, foram realizadas as descrições em VHDL dos componentes que compõem a arquitetura para prototipação do algoritmo de reação.
- h) Na fase 8 foram realizados testes na FPGA para verificação do funcionamento da arquitetura final proposta.
- i) Na última fase foi feita a documentação do trabalho.

1.5 ORGANIZAÇÃO DO TRABALHO

O trabalho está dividido em seis capítulos. O **capítulo 1** traz a introdução do estudo com informações essenciais, tais como problema, justificativa, objetivos, metodologia e sua organização. O **capítulo 2** apresenta o marco teórico do projeto, onde são abordados aspectos de tecnologias assistivas, filtragem Kalman, algoritmo de reação e *hardware* reconfigurável. O **capítulo 3** descreve o processo de calibração dos sensores, a proposta do algoritmo de reação após a fusão sensorial e as implementações em *software* e *hardware*. O **capítulo 4** mostra a metodologia e implementações adicionais necessária na etapa de testes na FPGA da arquitetura proposta. O **capítulo 5** apresenta os resultados de caracterização dos circuitos propostos, assim como a validação através de processos de simulação; e o **capítulo 6** apresenta as conclusões deste trabalho.

2 MARCO TEÓRICO

2.1 TECNOLOGIAS ASSISTIVAS VOLTADAS À LOCOMOÇÃO DE DEFICIENTES VISUAIS

Existem vários trabalhos voltados à construção de dispositivos para auxílio à locomoção de deficientes utilizando sensores e eletrônica embarcada para processamento. Destaca-se as bengalas eletrônicas, com sensores acoplados, geralmente ultrassom, microcontrolador para processamento de dados e motores vibratórios, cujo sistema completo é capaz de detectar obstáculos acima e abaixo da linha da cintura a cerca de 1 m de distância. Os preços variam entre 500 reais a 3,12 mil reais (Brentano, 2011). Com proposta semelhante, o americano Steve Hoefer criou o projeto da chamada luva-eletrônica, dispositivo que fica preso à mão do usuário, indicando a aproximação de obstáculos através da vibração de motores localizados acima da mão, com maior intensidade na medida em que o obstáculo se aproxima. O projeto é aberto e seu custo é de 60 dólares (Hoefer, 2011).

O robô-cão guia é outro dispositivo eletrônico criado para assistir a mobilidade de deficientes visuais. A empresa japonesa NSK apresenta uma solução de robô, simulando um cão, que através da utilização de sensores do Microsoft Kinect é capaz de detectar obstáculos e escadas. Seu sistema mecânico é capaz de subir escadas e o robô tem capacidade para atingir velocidades de até 3,7 km/h, quase a velocidade em que uma pessoa está caminhando. Possui também reconhecimento de voz do dono e um sistema de orientação baseado em GPS e mapas *online* capazes de serem programados pelo usuário, permitindo ao deficiente indicar a localidade em que deseja alcançar para o robô orientá-lo durante a caminhada. Entretanto, o produto ainda está em fase de testes, com expectativa de vendas a partir do ano de 2020 (Portal Terra, 2011). A Figura (2) mostra as tecnologias descritas anteriormente.

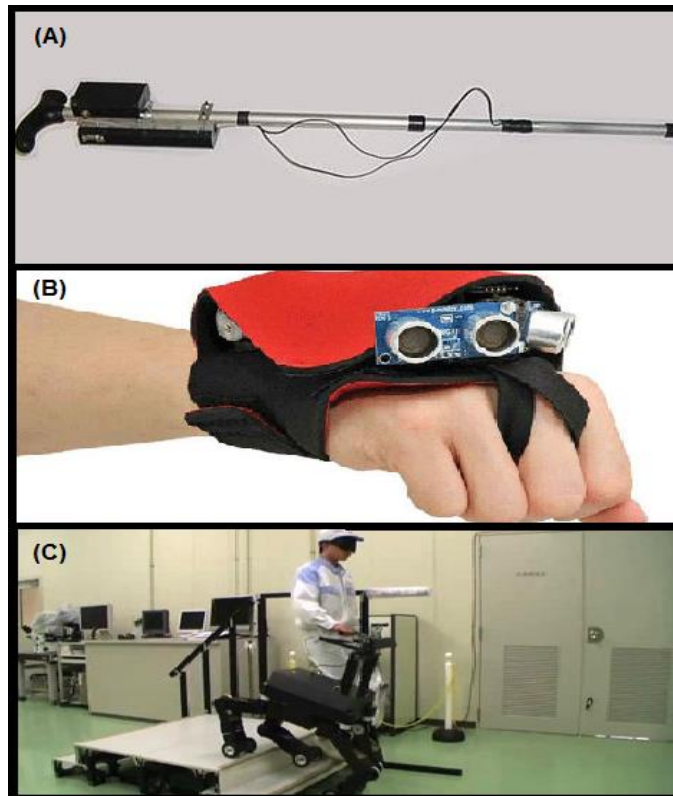


Figura 2. Tecnologias assistivas à locomoção de deficientes. (a) bengala eletrônica. (b) luva eletrônica (c) robô cão-guia.

2.2 FILTRO DE KALMAN

2.2.1 Introdução

Antes de apresentar o contexto histórico, faz-se necessário definir conceitos importantes para entender o funcionamento dos algoritmos do filtro de Kalman que serão apresentados neste trabalho. São eles: estimação, predição, filtragem, variável de estado e processo estocástico.

A estimação é definida como o processo de mensurar uma quantidade de interesse por meio de observações de medidas imprecisas e/ou incertas (Santos, s.d.).

A predição é o valor que uma estimativa no tempo presente terá em um intervalo de tempo futuro (Kim, 2011).

A filtragem corresponde ao processo de eliminação de sinais ruidosos indesejados durante a medição de uma determinada grandeza (Santos, s.d.).

O sistema estocástico é um acontecimento governado por leis físicas que, apesar da detenção de conhecimento de suas condições iniciais, se torna impossível prever seus estados futuros (Santos, s.d.).

A variável de estado é uma grandeza física utilizada para descrever um sistema estocástico.

Em 1960, Rudolph Emil Kalman publicou um artigo intitulado "Uma Nova Abordagem para Filtragem Linear e Problemas de Predição", no qual apresentava seu estudo em torno do problema da filtragem e predição clássica. Seus resultados mostravam a formulação de um novo modelo matemático recursivo para solucionar a filtragem de dados baseados no princípio de estimação e predição, além da separação de ruídos indesejados na medida observável de um sistema estocástico (Kim, 2011).

Esse método é comumente comparado a um filtro passa-baixas de primeira ordem, pois este é capaz de atenuar altas frequências e deixar passar as baixas frequências, sendo bastante utilizado para filtragem de sinais, uma vez que os ruídos associados à medida são considerados de alta frequência e o sinal que se quer mensurar possui baixa frequência. Apesar do comportamento similar, os dois possuem certas diferenças em relação à modelagem matemática, pois no filtro de primeira ordem os parâmetros que ajustam a medida são estáticos, ou seja, não variam no tempo. O FK, por ser um algoritmo recursivo, possui seus parâmetros variados ao longo do tempo, o que pode acelerar a convergência ao valor esperado. (Kim, 2011).

Desde então, o filtro de Kalman vem sendo objeto de estudos e pesquisas tanto em aplicações quanto no intuito de aprimorar e até mesmo modelar novas versões do algoritmo, a fim de se resolver problemas cada vez mais complexos, dada a sua vasta aplicação, principalmente na área de controle de sistemas dinâmicos, financeira e estatística (Aiube, 2005; Kalman, 1960). Para este trabalho, será utilizado o filtro de Kalman simples para a filtragem dos dados obtidos dos sensores infravermelho e ultrassom. Para a fusão dos dados e a consequente resposta que o sistema deve oferecer ao usuário, será usado o método de fusão sensorial. Nas próximas seções, são abordados os conceitos teóricos desses dois algoritmos.

2.2.2 Filtro de Kalman simples para filtragem de ruído

Nesta seção será abordado o filtro de Kalman simples para aplicação em filtragem de ruído. A explicação será constituída em quatro partes: a primeira apresentará o diagrama de fluxo do algoritmo e nomeará as variáveis presentes; a segunda explicará o processo de predição; a terceira elucidará o processo de estimação e a última parte apresentará a definição de parâmetros do sistema aplicada ao problema a ser resolvido por este trabalho. É importante salientar que a intenção da abordagem neste capítulo não é derivar as fórmulas matemáticas do filtro de Kalman, mas apenas apresentar o algoritmo e explicar o funcionamento de cada passo do seu processo de iteração.

2.2.2.1 Algoritmo do filtro de Kalman simples

A Fig. (3) a seguir apresenta o diagrama de fluxo do algoritmo do filtro de Kalman, com todos seus 5 passos.

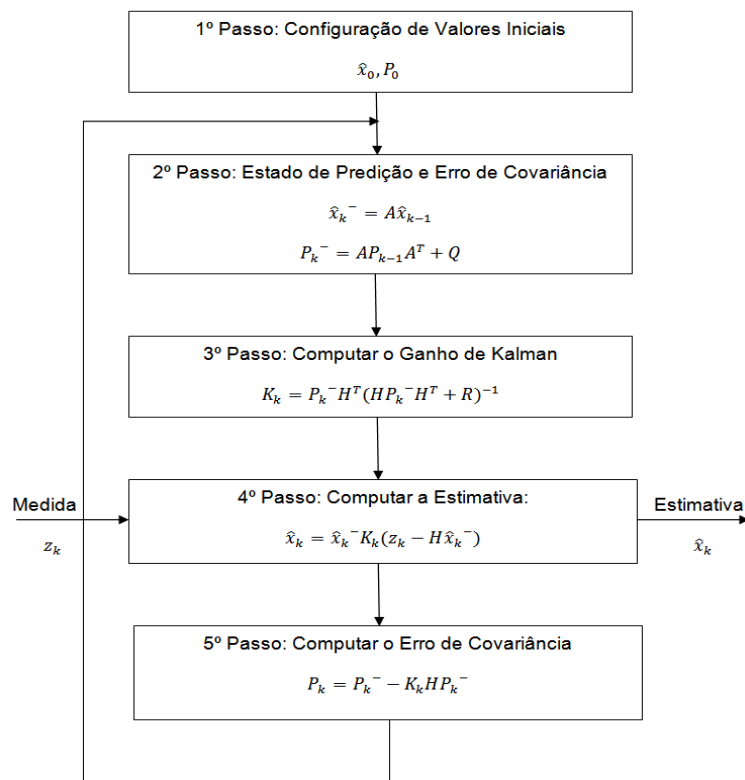


Figura 3. Fluxo de algoritmo do algoritmo do Filtro de Kalman (Kim, 2011).

Em seguida, serão nomeados e definidos cada parâmetro e variável envolvidos no fluxo, e também serão explicados, brevemente, os passos deste algoritmo, com maiores detalhes nos próximos subcapítulos.

- a) \hat{x}_0 : representa a estimativa inicial da variável de estado, ou seja, o valor real que a medida deveria assumir.
- b) P_0 : representa o erro de covariância inicial. O erro de covariância indica a diferença entre a estimativa do filtro de Kalman e o valor real da medida, embora seja desconhecido.
- c) \hat{x}_k^- e P_k^- : representam, respectivamente, os valores preditos da estimativa e erro de covariância. São diretamente ligados ao modelamento do sistema.
- d) K_k : representa o ganho de Kalman, uma variável que é responsável em ajustar os valores estimados da medida para cada passo se aproximar do valor real da medida e ajuste do erro de covariância.
- e) z_k : representa a medida lida por um instrumento de medição.
- f) \hat{x}_k : representa o valor estimado da medida.
- g) A , Q , R e H são matrizes que dependem do sistema a ser modelado para uso do filtro de Kalman e, portanto, não possuem seus valores alterados durante as iterações do algoritmo. “ A ” é nomeada como a matriz de transição de estados; “ Q ” é a matriz de covariância do ruído associado ao estado de transição; “ R ” é a matriz de covariância do ruído de medida e “ H ” é a matriz de covariância do ruído associado à medida instrumental.

2.2.2.2 Processo de predição

O processo de predição corresponde ao segundo passo do algoritmo do filtro de Kalman e é responsável por prever qual valor as estimativas \hat{x}_k e P_k terão no próximo intervalo de tempo $k + 1$. Nesta parte são consideradas as matrizes A e Q , que são as matrizes associadas à transição da variável de estado.

É importante ressaltar que o processo de predição é crucial para definir a performance do filtro de Kalman ao longo do tempo, uma vez que é usado para definir

o ganho de Kalman que influenciará na diferença entre o valor medido pelo sensor e o valor predito da medida naquele instante de tempo, conforme a equação mostrada no 4º passo.

2.2.2.3 Processo de estimação

Este processo corresponde ao terceiro, quarto e quinto passos do algoritmo da Fig. (3). Seu papel é computar a estimativa da medida, o produto final do filtro de Kalman.

O terceiro passo consiste em calcular o ganho de Kalman, considerando as matrizes H e R do modelamento do sistema, bem como o erro de covariância predito, que terá o papel de ajustar a diferença entre a medida instrumental e a medida computada usando a predição, somado à própria medida computada, como descrito no passo 4.

O passo 5, por sua vez, atualiza a estimativa do erro de covariância a ser usado na próxima iteração. Esse erro é importante para indicar a diferença entre a estimativa do filtro de Kalman e o valor real da medida até então desconhecido, ou seja, é o grau de acurácia da estimativa, cujo valor é diretamente proporcional ao erro da estimativa, isto é, quanto maior for P_k , maior é o erro associado e vice-versa (Kim, 2011).

2.2.2.4 Parâmetros e modelagem do sistema

A matriz A representa a matriz de transição de estados. Ela descreve as equações com que o sistema se move ou varia ao longo do tempo, ou seja, as equações dinâmicas do sistema.

A matriz H representa a matriz de estado para medida. Descreve a relação entre a medida e a variável de estado. No contexto das medidas sensoriais, descreve a precisão e acurácia do instrumento, ou seja, o quão próximo está o valor medido do valor real.

A matriz Q representa a matriz de covariância do ruído associado à variável de estado.

R é a matriz de covariância do ruído associado à medida do instrumento de medição.

Para fazer a modelagem de um sistema simples aplicado ao contexto deste trabalho, foi levado em consideração o modelo simples apresentado como caso de estudo em (Kim, 2011). Para obter os parâmetros necessários ao algoritmo do FK, faz-se necessário conhecer as fórmulas dadas pelas seguintes equações Eq. (1) e Eq. (2) apresentadas abaixo, usadas para modelagem de sistemas.

$$x_{k+1} = Ax_k + w_k \quad (1)$$

$$z_k = Hx_k + v_k \quad (2)$$

Onde x_k representa a variável de estado, z_k a medida instrumental, w_k o ruído associado à transição de estado, v_k o ruído associado à medida e A e H são as matrizes descritas acima.

No contexto do projeto, no qual o processo de obtenção de amostras se dá a uma taxa de amostragem muito mais rápida que a velocidade com que a pessoa caminha, pode-se assumir que não haverá mudanças bruscas do valor real de medição entre uma amostra e a seguinte, eliminando assim o ruído w_k . Assume-se também que cada sensor possui um ruído intrínseco associado. Sendo assim, a estimativa da medida dos sensores usados no projeto pode ser modelada pela Eq. (3) e a medida dos sensores é modelada pela Eq. (4) abaixo:

$$x_{k+1} = x_k \quad (3)$$

$$z_k = x_k + v_k \quad (4)$$

Q e R são matrizes de covariância que estão relacionadas a ruídos. Q relaciona-se com a variável w_k , e R relaciona-se com v_k . Ruídos são modelados estatisticamente por distribuições de probabilidade, pois não é possível ter precisão sobre o valor assumido e quando irão ocorrer.

De acordo com Kim (2011), tais matrizes são complicadas de serem definidas analiticamente, tendo seus valores determinados através de resultados experimentais e até mesmo por tentativa e erro.

Ademais, analisando-se as equações do 3º e 4º passos da Fig. (3), infere-se que a medida em que R aumenta o ganho de Kalman diminui, o que leva a um aumento da contribuição da estimativa no tempo anterior e diminuição da contribuição

da medida no cálculo da estimativa atual. Portanto, caso se queira diminuir a influência da medida na estimação da variável de estado e uma menor variação da estimativa, deve-se aumentar o valor de R .

Comparando-se as Eq. (3) e Eq. (4) com as Eq. (1) e Eq. (2), é possível determinar os valores para A , H e Q , como sendo 1, 1 e 0, respectivamente. Como não há informação a respeito da distribuição de probabilidade e da covariância dos ruídos associados aos sensores, definiu-se para R , por tentativa e erro, o valor de 4. Para a estimativa inicial da variável de estado, definiu-se 70 para simular o sistema a 70 cm de um obstáculo.

Durante os processos de calibração dos sensores, observou-se que o sensor infravermelho apresentava medidas mais ruidosas que o sensor ultrassom, então definiu-se para os erros de covariância inicial P_0 o valor de 4 para o sensor ultrassom e o valor de 10 para o sensor infravermelho. É importante ressaltar que tais valores de covariância inicial também foram baseados em tentativa e erro. Todos esses valores apresentados serão utilizados durante as simulações em *hardware* e *software*.

A Tabela (1) apresenta então todos os parâmetros do algoritmo do FK utilizados nas simulações realizadas no projeto.

Tabela 1. – Parâmetros do algoritmo do filtro de Kalman utilizados em simulações.

Parâmetro	Valor
A	1
H	1
Q	0
R	4
P_0 - ultrassom	4
P_0 - infravermelho	10

2.2.3 Fusão sensorial

Conforme a seção 2.2.2, a intenção deste subcapítulo consiste em explicar as fórmulas da fusão sensorial, suas variáveis e funcionamento, não havendo aqui a preocupação em derivar as equações matemáticas apresentadas.

A fusão sensorial é obtida por meio da extensão do FK para duas medidas instrumentais, significando a inserção de duas novas fontes de erro - ruído. Utilizando critérios estatísticos para otimização, aliados à técnica dos mínimos quadrados ponderados, foram apresentadas em (Conde et al., 2013) as Eq. (5) e Eq. (6):

$$\hat{q} = q_1 + \frac{\sigma_k^2}{\sigma_k^2 + \sigma_z^2} (q_2 - q_1) \quad (5)$$

$$\sigma_{k+1}^2 = \sigma_k^2 - \frac{\sigma_k^2}{\sigma_k^2 + \sigma_z^2} \sigma_k^2 \quad (6)$$

As variáveis das equações são nomeadas e explicadas como se segue:

- a) \hat{q} é a estimativa da fusão: variável que representa a estimativa da fusão de dois sensores sob as mesmas condições de contorno, medindo a mesma variável de estado.
- b) q_1 é a medida do sensor 1: medida realizada por um sensor denotado de 1.
- c) q_2 é a medida do sensor 2: medida realizada por outro sensor, denotado de 2.
- d) σ_k^2 é o erro de covariância associado ao sensor 1.
- e) σ_z^2 é o erro de covariância associado ao sensor 2.
- f) σ_{k+1}^2 é o erro de covariância estimado do algoritmo de fusão sensorial no instante de tempo $k + 1$.

Comparando-se as Eq. (5) e Eq. (6) com as equações apresentadas nos passos 4 e 5 da Fig. (3), pode-se afirmar que tal algoritmo de fusão sensorial é baseado em duas equações do algoritmo de FK, sendo que o ganho de Kalman, denotado por K_{k+1} é dado pela Eq. (7).

$$K_{k+1} = \frac{\sigma_k^2}{\sigma_k^2 + \sigma_z^2} \quad (7)$$

Analisando as equações apresentadas para o algoritmo de fusão sensorial, é possível concluir que a medida realizada pelo sensor 1 é a que possui maior peso para a estimativa da medida de fusão sensorial e cálculo do erro de covariância no instante futuro. Espera-se, portanto, que os resultados aproximados para a estimativa da fusão sensorial sejam mais próximos dos valores medidos pelo sensor 1 do que pelo sensor 2.

A escolha do sensor 1 depende do modelo do sistema. Para este trabalho, foi escolhido o sensor ultrassom, por possuir menor ruído e comportamento linear. Os erros de covariância dos sensores foram mantidos do algoritmo do filtro de Kalman simples descrito na Seção 2.2.2. Assim sendo, σ_0^2 para o sensor ultrassom e infravermelho são, respectivamente, 4 e 10.

A Tabela (2) apresenta então todos os parâmetros do algoritmo de fusão sensorial utilizados nas implementações realizadas no projeto.

Tabela 2. – Parâmetros do algoritmo de fusão sensorial utilizados nas simulações.

Parâmetro	Valor
σ_0^2 - ultrassom	4
σ_0^2 - infravermelho	10

2.3 ARQUITETURAS RECONFIGURÁVEIS - FPGA

Dispositivos “*Field Programmable Gate Arrays*” (FPGAs) são circuitos integrados reconfiguráveis e programáveis em linguagens de descrição de *hardware* (HDLs - *Hardware Description Languages*), como Verilog ou VHDL - *VHSIC Hardware Description Language* -, usados para prototipagem de circuitos integrados de aplicação específica (ASICs).

Desenvolvidas pelos primeiros co-fundadores da empresa Xilinx Inc, Ross Freeman e Bernard Vonderschmitt em 1985 (National Instruments, 2013), as FPGA são compostas de uma quantidade finita de recursos de *hardware*, tais como:

- a) Blocos lógicos configuráveis, formados basicamente por *Flip-Flops (FFs)* - bloco de memória de um bit controlado por um relógio (*clock*) - e *Look-Up Tables (LUTs)* - dispositivos que permitem implementar tabelas verdade de uma função lógica booleana.
- b) Blocos de memória *Random Access Memory (RAM)*.
- c) Blocos de funções lógicas definidas, como multiplicadores e blocos de processamento de sinais digitais – *Digital Signal Processing (DSPs)* – processadores constituídos de multiplicadores e somadores integrados, permitindo poupar o uso de LUTs e FFs.

Existem também outros recursos como os *Phased Locked Loop* (PLL), e *Data Clock Management* (DCMs) que permitem maior flexibilidade para implementações em *hardware*.

Dentre suas diversas vantagens, destaca-se a capacidade de paralelismo, ou seja, FPGAs são capazes de realizar diferentes operações de processamento ao mesmo tempo, por não consumirem os mesmos recursos de *hardware*. Tarefas independentes possuem seções de chip dedicadas funcionando de maneira autônoma, permitindo alto desempenho.

A Figura (4) mostra um kit de desenvolvimento FPGA Nexys4, o qual contém um chip FPGA da família Artix7 da Xilinx, além de diversas e interfaces de comunicação com periféricos externos.

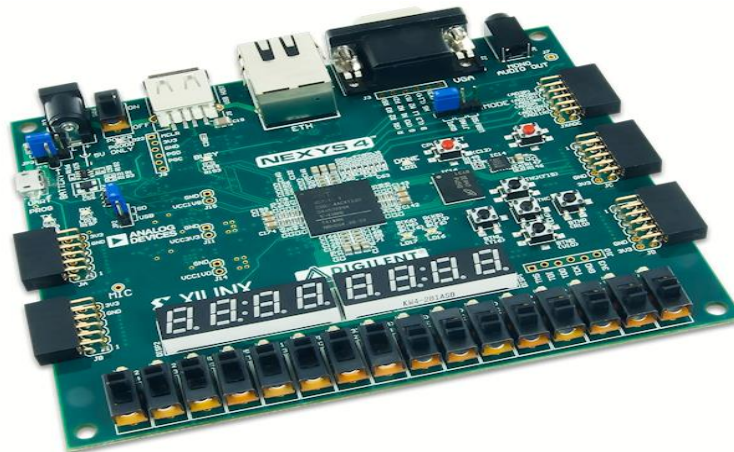


Figura 4. FPGA Nexys 4 (Digilent).

2.3.1 Fluxo de projeto em FPGAs

O fluxo de projeto em FPGAs é mostrado na Fig. 5. É dividido nas etapas descritas a seguir (Kilts, 2007):

- a) Especificação funcional: nesta etapa é realizada a análise de requisitos de projeto em *hardware*.
- b) Descrição de *hardware* e otimização: nesta etapa é realizada a descrição de *hardware* baseada nas HDLs e, em seguida, a otimização da descrição previamente realizada.

- c) Simulação comportamental: nesta etapa é analisado o comportamento da descrição realizada.
- d) Síntese lógica: após a etapa anterior ser implementada com sucesso, é feita a síntese da descrição realizada em forma de funções lógicas booleanas.
- e) *Place and route*: é feito o mapeamento e conexão das funções lógicas sintetizadas nos elementos que compõem a FPGA.
- f) Etapa de simulação funcional ou elétrica: processo de verificação de requisitos de *timing* e consumo de potência da arquitetura desenvolvida.
- g) Prototipagem em FPGAs: após a implementação correta das etapas anteriores, faz-se a prototipagem na FPGAs para obter resultados como consumo de recursos de *hardware* da FPGA e de recursos de circuitos como frequência de operação e consumo de potência.

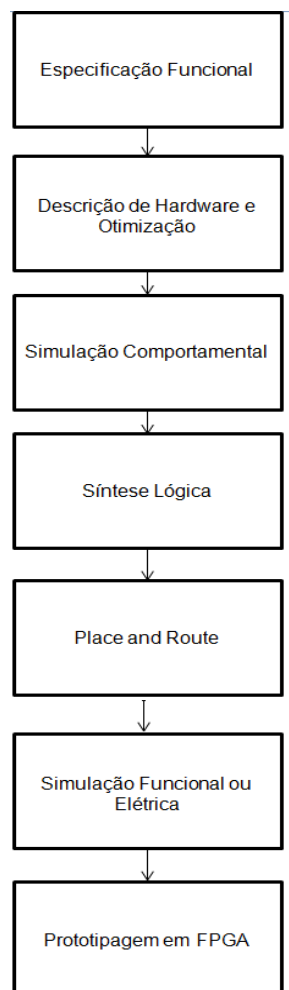


Figura 5. Fluxo de projeto de FPGAs, adaptado de (Kilts, 2007).

3 IMPLEMENTAÇÕES

3.1 ALGORITMO DE REAÇÃO

Nesta seção é apresentada a proposta de um modelo para o algoritmo de reação aos valores calculados na etapa de fusão sensorial. Esse algoritmo tem como objetivo informar ao usuário qual a direção que ele deve seguir quando se detecta um obstáculo por meio de vibração de atuadores piezoelétricos.

Primeiramente foi realizado o estudo sobre o algoritmo de zonas virtuais deformáveis (ZVD) como proposta para o algoritmo de reação. Entretanto, após realizada a análise de como é feito o modelamento, detalhados em Paim (2005), constatou-se que os cálculos são complexos, envolvendo equações dinâmicas e derivadas parciais, e, portanto, devido à complexidade, descartou-se o algoritmo ZVD como proposta de modelo de reação aos obstáculos.

Após descartada a utilização do algoritmo ZVD, foi proposto um modelo simples, baseado em uma zona de proteção circular imaginária de reação vetorial. Quando essa zona de raio fixo, medido em cm, engloba um obstáculo, detectado por um sensor, o algoritmo reage para direcionar o usuário a fim de se retirar o obstáculo dessa zona.

Considerou-se para esse algoritmo que o usuário estará andando para a frente ou para os lados, nunca em diagonal, e estará em uma sala com paredes laterais perpendiculares e parede frontal paralela ao usuário, como mostra a Fig. (6). Além disso, considera-se também que o número máximo de obstáculos detectados é igual a dois, sendo que só pode ser lido um obstáculo nos sensores laterais (todos exceto o par de sensor 4, que representa o sensor frontal), e/ou um obstáculo no sensor frontal.

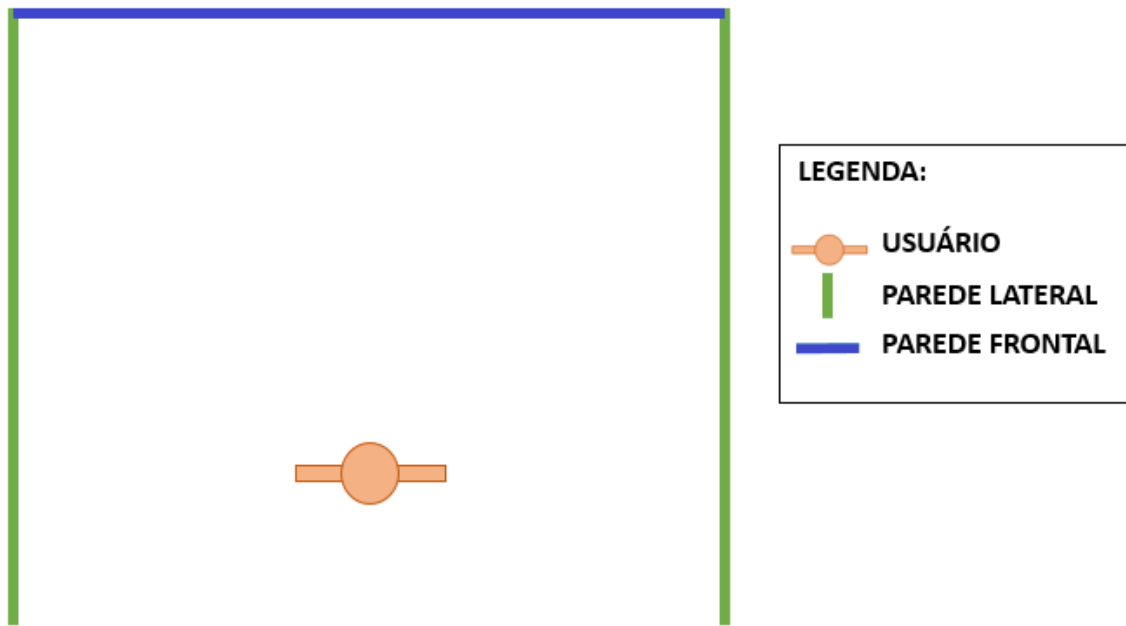


Figura 6. Considerações para simulações: disposição das paredes.

Conforme visto na Fig. (1) e na introdução deste trabalho, o dispositivo é utilizado pelo usuário na cabeça, de formato semicircular. Realizou-se um espaçamento uniforme entre os sete pares de sensores e, portanto, da direita para a esquerda, os sensores são numerados de 1 a 7 e posicionados nos ângulos 0°, 30°, 60°, 90°, 120°, 150° e 180° e, além disso, em cada lado (direito e esquerdo) do dispositivo, existem dois atuadores piezoelétricos posicionados próximos às orelhas. O piezoelétrico posicionado à frente da orelha representa a reação que o usuário deve tomar para os lados (direita ou esquerda) e o piezoelétrico localizado atrás da orelha representa a reação pra se mover para trás.

A leitura de um par de sensor é representada por um vetor, cujo módulo (ou intensidade) representa a distância entre o sensor e o obstáculo. Assim, se um obstáculo é detectado dentro da zona imaginária, o vetor de reação deve ter sua intensidade igual ao valor do raio de proteção menos a distância lida pelo par de sensor, com direção contrária à direção do sensor que detectou o obstáculo, como visto na Eq. (8).

$$MOD_{VR} = R - |S_X| \quad (8)$$

As variáveis das equações são nomeadas e explicadas a seguir:

- a) MOD_{VR} é o módulo do vetor resultante.
- b) R é o raio de proteção definido para o usuário.
- c) $|S_X|$ é o módulo do valor lido por um sensor, ou seja, a distância para um obstáculo.

Pela Eq. (8) acima, percebe-se que quanto mais perto da zona de proteção estiver o obstáculo, menor será a intensidade do vetor de reação e quanto mais perto da pessoa, maior será a intensidade. Para facilitar a implementação, o vetor de reação foi decomposto em dois vetores, nas direções x e y do plano cartesiano em um sistema bidimensional, assim como os sensores laterais posicionados nos ângulos de 30° , 60° , 120° e 150° foram decompostos apenas na direção x , logo o módulo do vetor decomposto é dado pelo cosseno do ângulo multiplicado pelo valor lido pelo sensor.

A vibração de um atuador piezoelétrico pode ser controlada por um pulso PWM. Quanto maior for o *duty cycle* (fração do período da onda que se mantém em estado ativo), mais forte será a vibração e vice-versa. Assim sendo, o algoritmo produz na sua saída um valor que deve alterar o *duty cycle* do PWM, conforme a necessidade para desvio do obstáculo.

Na Tabela (3) é apresentada a lógica para acionamento dos piezoelétricos quando um obstáculo está dentro da zona de proteção. Caso seja detectado um obstáculo pelos sensores laterais, liga-se o atuador oposto, indicando a direção que deve ser tomada pelo usuário. Caso seja detectado um obstáculo pelo par de sensores frontal, liga-se os dois atuadores traseiros, se não houver nenhum obstáculo detectado pelos sensores laterais, caso haja, será ligado apenas o atuador na direção oposta.

A intensidade de vibração é definida através do resultado da Eq. (8), que será posicionada em 3 níveis: fraco, média e forte. Para uma zona de proteção de x cm de raio, sendo $x > 20$, os níveis são divididos em:

- a) 0 a 10 cm: nível fraco.
- b) 10 a 20 cm: nível médio.
- c) 20 a x cm: nível forte.

Tabela 3. – Acionamentos dos atuadores quando um sensor detecta um obstáculo: D representa desligado e L representa ligado.

SENSORES	ATUADORES PIEZOELÉTRICOS			
	Dianteiro direito	Traseiro direito	Dianteiro esquerdo	Traseiro esquerdo
Direito (s1 a s3)	D	D	L	D
Esquerdo (s5 a s7)	L	D	D	D
Frontal (s4)	D	L	D	L
Frontal e direito (s1 a s4)	D	D	L	L
Frontal e esquerdo (s1 a s4)	L	L	D	D

No algoritmo de reação foi implementada também uma função adicional para levar o usuário para o meio do ambiente na direção x. Para isso, essa função desabilita a detecção de obstáculos na direção x (considera que o local que o usuário se encontra tenha obstáculos apenas na direção y) e considera apenas as medidas entre dos pares de sensores $s1$ e $s7$, com alcance máximo de 150 cm, para as paredes laterais do local em que se encontra. Sendo assim, o módulo do vetor resultante na direção x é dado pela Eq. (9) abaixo.

$$MOD_{VR} = \frac{-(s1-s7)}{2}, \text{ se } s1 > s7; \frac{(s7-s1)}{2} \text{ se } s7 > s1 \quad (9)$$

As variáveis das equações são nomeadas e explicadas a seguir:

- a) MOD_{VR} é o módulo do vetor resultante.
- b) $s1$ é o valor de distância de uma parede lido pelo par de sensores 1 somado ao raio do dispositivo de formato circular da Fig. (1).
- c) $s7$ é o valor de distância de uma parede lido pelo par de sensores 7 somado ao raio do dispositivo de formato circular da Fig. (1).

3.2 ASPECTOS GERAIS

Nesta subseção é feita uma descrição dos recursos de *hardware* e *software* usados para o desenvolvimento deste trabalho. Em seguida, é apresentada uma

descrição metodológica dos processos de calibração dos sensores, assim como da caracterização dos circuitos propostos. Finalmente são mostradas as arquiteturas propostas para a implementação em dispositivos reconfiguráveis dos Filtros Kalman e do algoritmo de reação.

3.2.1 Ambientes e programas para implementações de *software*

Nas implementações realizadas no TCC 1, para as simulações em *software*, foi utilizado um computador com sistema operacional Windows 7 Home Premium de 64 bits, com processador Intel® Core™ i7, modelo 3612QM, de frequência de operação de 2,10 GHz e memória RAM instalada de 8,00 GB. Os programas foram desenvolvidos em linguagem C e compilados com o GCC no terminal do Windows.

Para obtenção do modelo matemático dos sensores ultrassom e infravermelho a partir de dados do processo de calibração, além dos modelos do filtro de Kalman e fusão sensorial, utilizou-se o software MATLAB versão R2012a juntamente com seu recurso de obtenção de equações a partir de interpolação gráfica, chamado *Basic Fitting*.

Para a analisar o desempenho dos algoritmos, foi utilizada a ferramenta *Profile Analysis*, disponível pelo compilador GCC e executada no programa DEV C++, permitindo analisar as funções de uma aplicação de software e seus respectivos tempos de execução.

Nas implementações realizadas no TCC 2, para as simulações de software, foi utilizado o sistema operacional Ubuntu 14.04 LTS DE 64 bits, com processador Intel® Core™ i7, modelo 3612QM, de frequência de operação de 2,10 GHz e memória RAM instalada de 8,00 GB.

Utilizou-se ainda o software MATLAB versão R2013a para simular o algoritmo de reação e para comparação dos resultados de software e hardware.

3.2.2 Ambiente e programa para implementações de Hardware

No TCC1, para as implementações de *hardware*, foi utilizado o ambiente de desenvolvimento de projetos de descrição de circuitos *Xilinx ISE Design Suite*, versão

12.1. Os programas foram desenvolvidos na linguagem VHDL e simulados com o ambiente ISim, integrado no ambiente de desenvolvimento. Já no TCC 2, utilizou-se a versão 14.3 do programa *Xilinx ISE Design Suite*.

Resultados de síntese, como número blocos lógicos configuráveis (CLBs), registradores *Flip-Flops (FFs)* e *Look-Up Tables (LUTs)*, blocos DSPs, bem como resultados da frequência máxima de operação do circuito são fornecidos após a síntese lógica dos circuitos.

Fez-se também o uso de blocos de propriedade intelectual (*IP cores*) desenvolvidas pelo Prof. Daniel Muñoz para implementação de operações matemáticas de adição, subtração, multiplicação e divisão em ponto flutuante usadas nos algoritmos.

3.2.3 Sensor ultrassom

O sensor ultrassom é um dispositivo que é capaz de medir distâncias entre um objeto e ele próprio através da medição do tempo do processo de emissão de ondas sonoras na frequência de aproximadamente 20kHz que saem do sensor, atingem o alvo e retornam ao receptor (Morris, 2001). Foi escolhido para esse projeto o sensor ultrassom LV-MaxSonar-EZ0, da empresa MaxBotix, com tensão de alimentação entre 2,5 e 5,5 V e alcance de medição entre 0 a 6,45 m. Sua resposta por polegada é definida pela divisão entre a tensão de alimentação por 512. Assim sendo a curva de resposta do sensor é linear e crescente. A Figura (7) mostra o sensor ultrassom utilizado neste projeto. Seus benefícios incluem:

- a) Baixo custo.
- b) Grande alcance.
- c) Baixo consumo de potência - corrente de alimentação de 2 mA.
- d) Taxa de amostragem de aproximadamente 20Hz.
- e) Saídas analógica e digital (Maxsonar LV-EZ1-MB1010, 2005).



Figura 7. Sensor ultrassom (Maxsonar LV-EZ1-MB1010, 2005).

3.2.4 Sensor infravermelho PSD

O sensor infravermelho *Position Sensing Detector* (PSD) é composto por um emissor, um led infravermelho, receptor de luz, um fotodiodo, que converte luz em tensão, e um circuito para processamento de sinal (Naiff et al., 2011; GP2Y0A02YK, 2001). A partir desses dois componentes, ele é capaz de medir distâncias entre si e um objeto a sua frente. Para este trabalho, foi escolhido o sensor infravermelho modelo GP2Y0A02YK0F, da fabricante Sharp, com tensão de alimentação entre 4,5 e 5,5 V, corrente de alimentação típica de 33 mA, com alcance de medição entre 20 a 150 cm. Sua saída é analógica e a resposta do instrumento à distância apresenta comportamento exponencial decrescente, conforme mostra a Fig. (8).

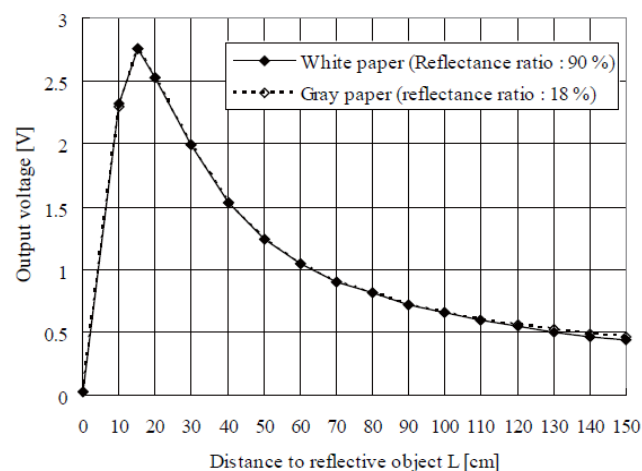


Figura 8. Curva de resposta do sensor infravermelho em função da distância (GP2Y0A02YK, 2001).

A Figura (9) mostra o sensor infravermelho usado no desenvolvimento deste trabalho.



Figura 9. Sensor infravermelho (Robotics-electronics).

3.2.5 Kit FPGA

Foi escolhida a placa de desenvolvimento Nexys™4 da Digilent Inc., com FPGA integrada Artix-7™ da fabricante Xilinx Inc. (vide Fig. (4)), por disponibilizar de:

- a) 15850 blocos lógicos, cada uma com quatro LUTs de 6 entradas e 8 FFs.
- b) 240 DSPs.
- c) Conversor ADC de 12 bits integrado no chip FPGA.

3.3 CALIBRAÇÃO DOS SENSORES

Calibração é um processo que consiste em comparar valores medidos de uma mesma grandeza por um instrumento ou sensor com valores medidos por outro instrumento com acurácia conhecida (Morris, 2001). Dessa forma, fez-se necessário para este trabalho a calibração dos sensores apresentados, de maneira a determinar seu comportamento e encontrar sua representação matemática.

Para tal, utilizou-se o conversor ADC de 10 bits – 1024 valores possíveis de leitura da porta analógica – do microcontrolador Arduino Mega para converter as tensões que os dois sensores produziam a partir das distâncias de um objeto em informação numérica e, assim, determinar o seu comportamento através de um gráfico com o auxílio do software MATLAB.

O processo de medição ocorreu da seguinte forma: posicionou-se os dois sensores, como mostrado na Fig. (10), em cima do marco 0 cm de uma fita métrica e foi-se deslocando, a partir do marco 20 cm, em 5 em 5 cm, uma caixa de madeira com 30 cm comprimento e 20 cm de altura (Fig. 11). A cada deslocamento da caixa foram medidos 10 valores no programa arduino elaborado para a leitura dos valores lidos pelo conversor ADC.

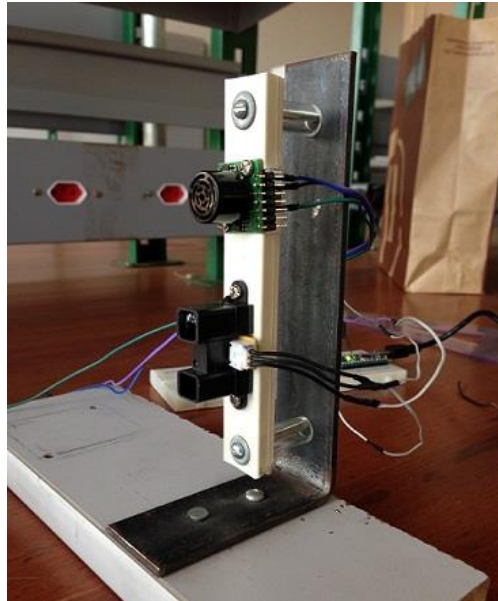


Figura 10. Disposição dos sensores para a calibração (acima se encontra o sensor ultrassom e abaixo o infravermelho).

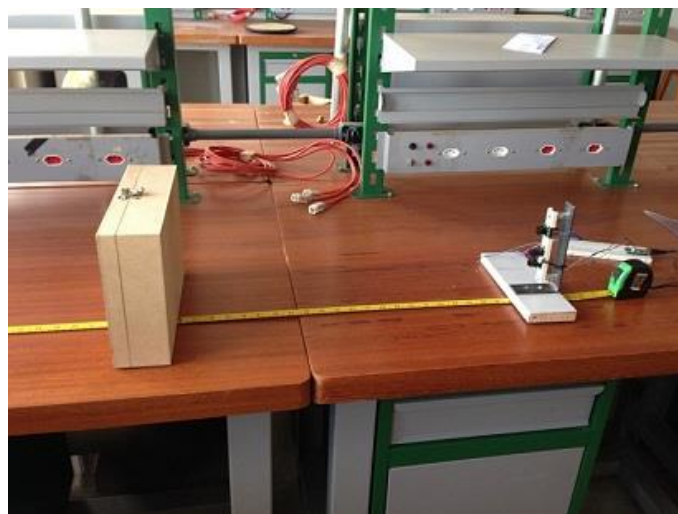


Figura 11. Calibração dos sensores.

Com os dados obtidos, fez-se a média dos 10 valores medidos a cada deslocamento da caixa para representar o valor final lido pelo ADC a cada distância medida, e então plotou-se os gráficos dos sensores (Medida do Conversor ADC X Distância da fita métrica). Após a confecção dos gráficos, com o comando *Basic Fitting*, encontrou-se a curva interpoladora dos dois sensores para obtenção da equação característica de cada sensor.

As curvas interpoladas dos sensores são apresentadas nas Fig. (12) e Fig. (13), bem como as equações Eq. (10) e Eq. (11) características do sensor ultrassom e infravermelho, respectivamente.

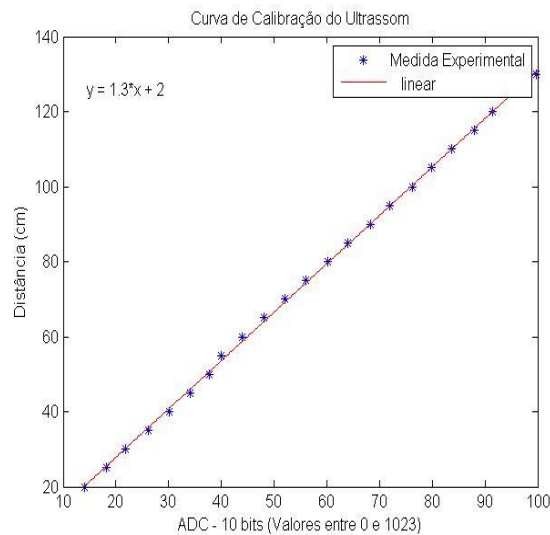


Figura 12. Curva de calibração do sensor ultrassom e modelo matemático.

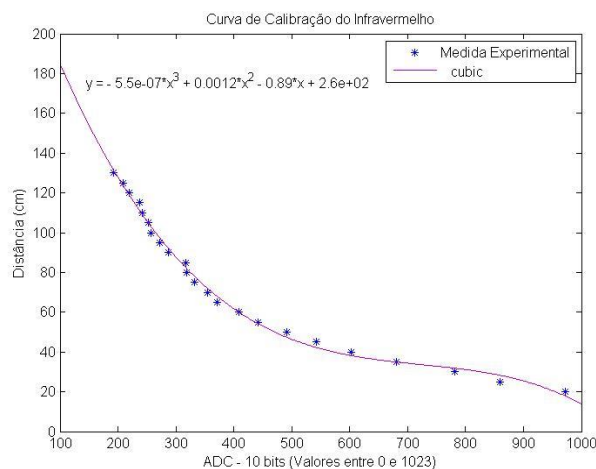


Figura 13. Curva de calibração do sensor infravermelho e modelo matemático.

As equações características são:

$$y = 1,3x + 2 \quad (10)$$

$$y = -5,5 \cdot 10^{-7}x^3 + 0,0012x^2 - 0,89x + 260 \quad (11)$$

3.4 IMPLEMENTAÇÕES EM HARDWARE E SOFTWARE

3.4.1 Implementações em *software*

3.4.1.1 Implementações em linguagem C

As implementações dos algoritmos de FK e fusão sensorial em linguagem C tinham como objetivo analisar o tempo que cada passo dos algoritmos levava em sua execução, para que fosse determinado se o sistema embarcado necessitaria de arquiteturas de *hardware* dedicadas para aceleração de cálculos, caso o tempo de execução fosse alto, ou se poderiam ser utilizadas apenas arquiteturas híbridas *hardware* e *software*.

Para isso, foram criados dois arquivos: em um estavam os parâmetros dos algoritmos e no outro estavam 9 amostras simulando as leituras dos sensores já convertidas para valores da distância em cm. Com o auxílio da ferramenta de *Análise de Perfil* do programa DEV C++, estimou-se o tempo computacional que cada passo da iteração completa demandava. Este processo foi repetido 200000 vezes em um loop, visando calcular uma média do tempo de execução.

São apresentadas nas Tab. (4) e Tab. (5) os resultados de análise de *profile* dos algoritmos fusão sensorial e FK, respectivamente, obtidos com o programa DEV C++.

Tabela 4. – Tempo de execução dos passos do algoritmo de fusão sensorial.

Função	Tempo (%)
Passo 1	40
Passo 2	20
Passo 3	0

Tabela 5. – Tempo de execução dos passos do algoritmo do Filtro de Kalman.

Função	Tempo (%)
Passo 1	0
Passo 2	28.57
Passo 3	14.29
Passo 4	0
Passo 5	0

Pode-se deduzir, observando a coluna Tempo (%) da análise de *profile*, que os passos 3 e o passo 1 dos algoritmos de FK e fusão sensorial, respectivamente, representam os processos que mais consomem tempo de execução dentro de todo o algoritmo. Entretanto, foram necessárias diversas chamadas das funções para que o *software* fosse capaz de fornecer as porcentagens de tempo apresentadas. Contudo, visando evitar atrasos pelas transferências de dados entre coprocessadores de hardware e trechos implementados em software, decidiu-se implementar todas as equações contidas nos algoritmos na arquitetura de *hardware*, sem a necessidade de auxílio de arquiteturas de *software* embarcadas.

3.4.1.2 Implementações em MATLAB

Foram gerados no MATLAB os gráficos obtidos pela simulação do FK e fusão sensorial, a fim de permitir uma melhor visualização do comportamento desses algoritmos ao longo do tempo. Utilizou-se os parâmetros discutidos no capítulo 2 deste trabalho.

São apresentadas na Fig. (14) e Fig. (15) os resultados da filtragem de Kalman para os sensores ultrassom e infravermelho através de 100 amostras aleatórias em intervalo de 0.2 segundos para valores de distância em cm que cada um fornecia. Na Figura (16) é apresentado o resultado da fusão sensorial utilizando as saídas filtradas obtidas nas Fig. (14) e Fig. (15).

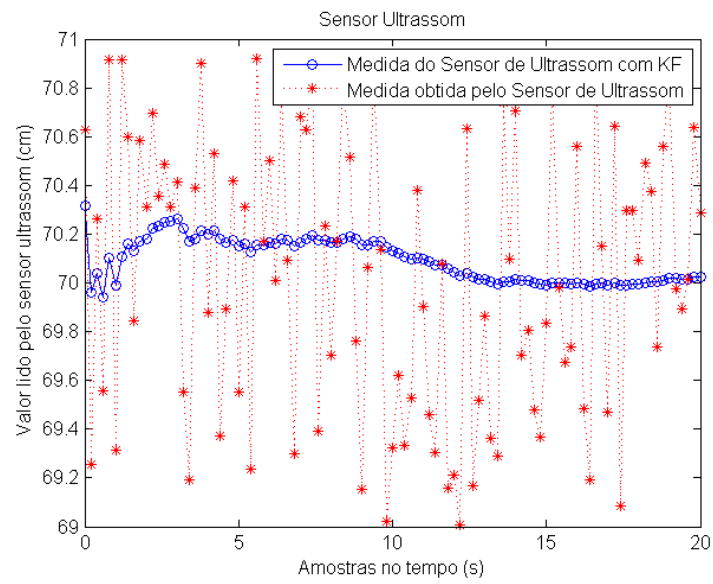


Figura 14. Resultado da implementação do FK para o sensor ultrassom.

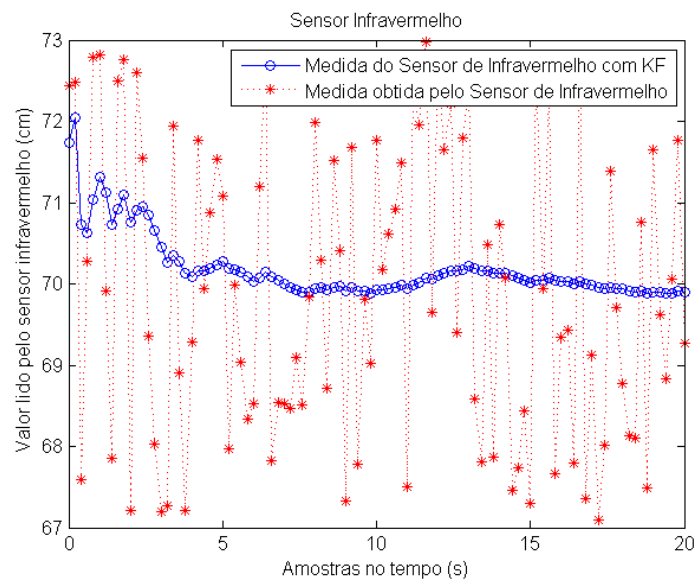


Figura 15. Resultado da implementação do FK para o sensor infravermelho.

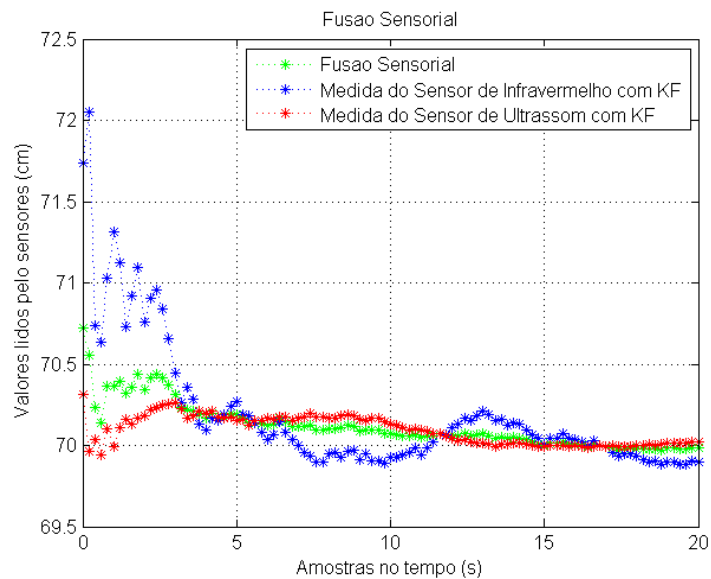


Figura 16. Resultado da fusão sensorial.

Os valores de teste foram gerados de forma aleatória com distribuição uniforme, de maneira a produzir 100 amostras de valores simulando as leituras dos sensores, tanto em forma de leitura do conversor analógico-digital, quanto em valores já convertidos para distância em cm.

Foram gerados também vetores de teste em binário utilizados durante a simulação comportamental de todos os blocos da arquitetura de *hardware* proposta. Foram criados programas que são capazes de avaliar o erro entre os resultados de arquitetura de *hardware* e de simulações em MATLAB para determinar a precisão dos resultados e a validação da arquitetura.

Para o algoritmo de reação, foi implementada uma função no MATLAB responsável por fazer uma animação contendo duas áreas gráficas, com o gráfico à esquerda mostrando os valores lidos pelos sensores e o outro gráfico mostrando o vetor de reação, conforme apresentado na Fig. (17), onde a região em verde indica a cabeça do usuário, os vetores em azul representam o módulo da distância a obstáculo lido pelo sensor e a região em vermelho representa a zona de proteção.

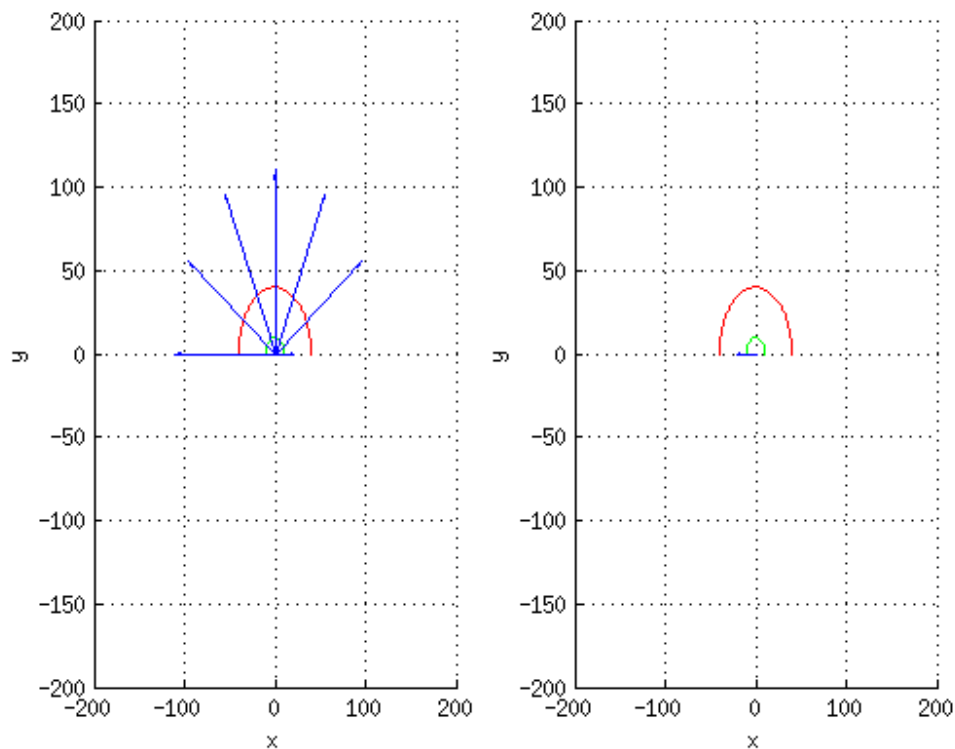


Figura 17. Algoritmo implementado em MATLAB para verificação do funcionamento do algoritmo de reação.

A animação consistia em simular o usuário caminhando em direção a um obstáculo à sua direita e depois se desviando.

3.4.2 Implementações em *hardware*

3.4.2.1 Arquitetura da fusão sensorial

Na Figura (18) é apresentada a solução para a arquitetura geral para a fusão sensorial de um par de sensores ultrassom e infravermelho. O sistema deve ser capaz de receber os dados provenientes de um conversor analógico-digital, e convertê-los para medidas de distância dos obstáculos em seu alcance, por meio dos modelos matemáticos obtidos durante o processo de calibração. Finalmente, as informações ruidosas dos sensores necessitam passar pelo processo de filtragem do FK, para que possam ser atenuados os ruídos e realizado o processo de fusão.

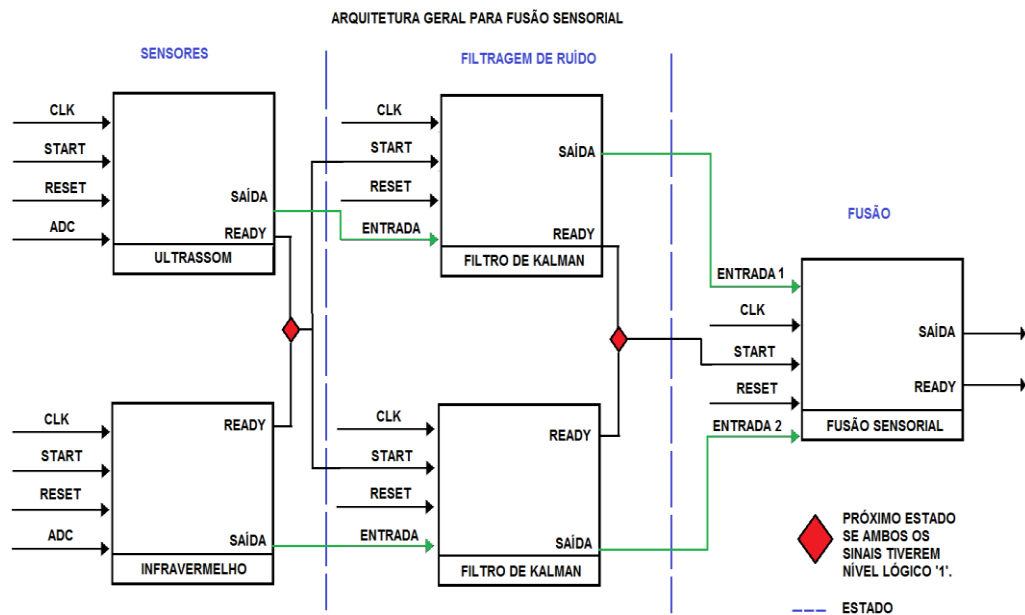


Figura 18. Arquitetura para fusão sensorial em diagrama de blocos.

Em seguida, são apresentadas as arquiteturas de *hardware* propostas para a implementação dos blocos da arquitetura geral para fusão sensorial. Todas foram descritas por meio de uma máquina de estados finitas. Os operadores matemáticos usados em cada estado são implementados em paralelo e compartilhados nos demais estados.

3.4.2.1.1 Modelo do sensor ultrassom

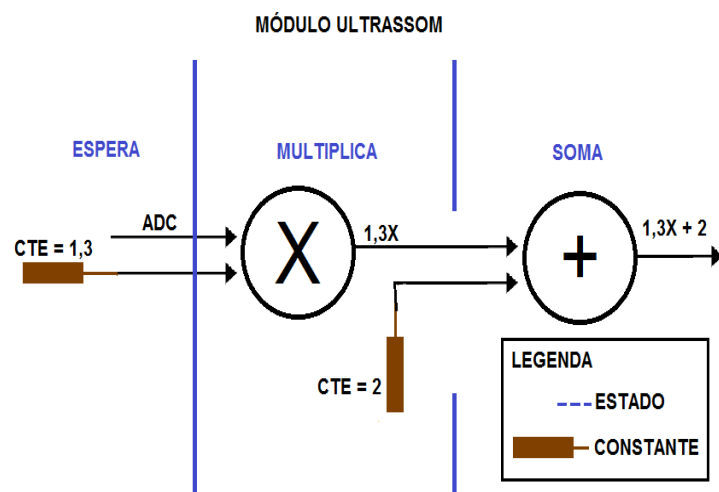


Figura 19. Arquitetura do modelo do sensor ultrassom.

3.4.2.1.2 Modelo do sensor infravermelho

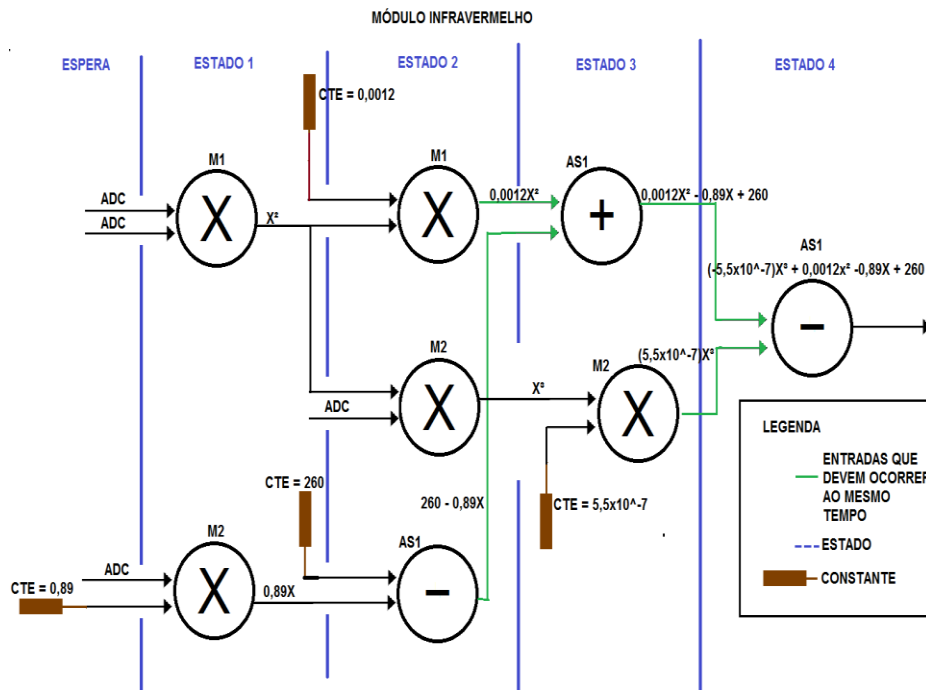


Figura 20. Arquitetura do modelo do sensor infravermelho.

3.4.2.1.3 Modelo do filtro de Kalman

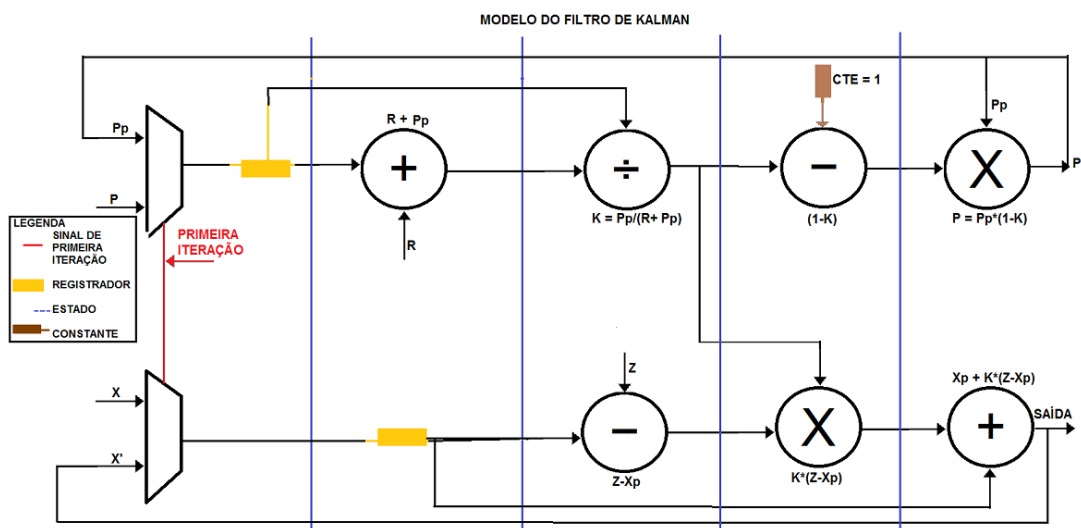


Figura 21. Arquitetura do algoritmo do filtro de Kalman.

3.4.2.1.4 Modelo da fusão sensorial

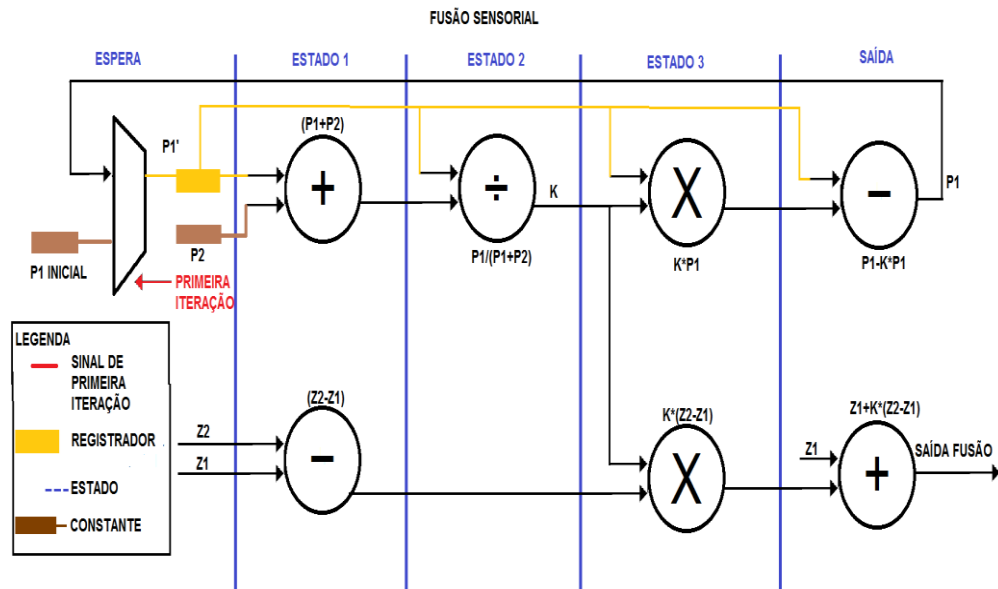


Figura 22. Arquitetura do algoritmo de fusão sensorial.

3.4.2.2 Arquitetura do algoritmo de reação

Para a implementação do algoritmo de reação, optou-se por dividir este módulo em três submódulos: o primeiro implementa o algoritmo descrito na seção 3.1, o segundo implementa um comparador para definir o *duty cycle* do pulso PWM, bem como os sinais de acionamento e *reset* para o último submódulo, que é o gerador do pulso PWM (Anexo I).

3.4.2.2.1 Diagrama de estados do módulo do algoritmo de reação

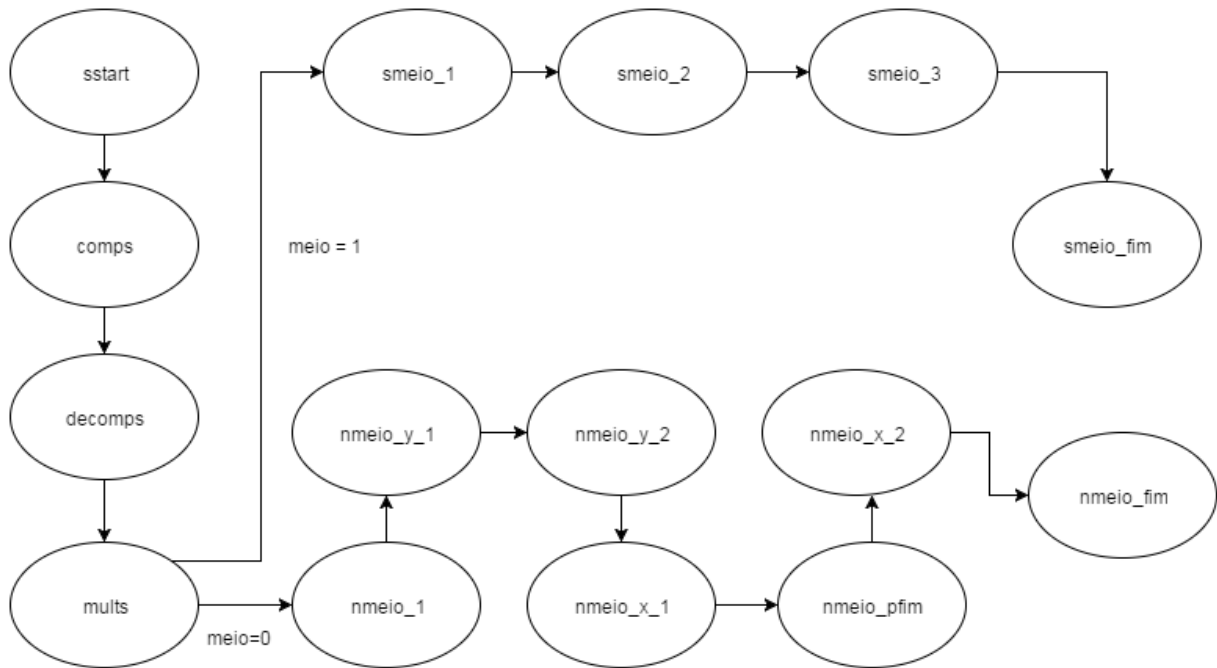


Figura 23. Diagrama de estados do algoritmo de reação.

Esse módulo implementa o algoritmo para cálculo dos módulos vetores de reação x e y . Recebe os valores fornecidos pelo bloco de fusão sensorial e um sinal para acionar a função de ir para o meio do ambiente. Cada estado possui as funções descritas a seguir.

No estado inicial *sstart*, são inicializados com o valor “0” todos os sinais utilizados na arquitetura, que aguarda o sinal de inicialização do bloco.

No estado *comps* são feitas comparações para detectar se há obstáculo dentro da zona, através da comparação do valor proveniente do bloco de fusão sensorial com o valor do raio da zona de proteção.

No estado *decomp* é realizada a decomposição em cosseno, caso exista obstáculo e a função meio esteja desligada, do valor lido por um dos sensores nos ângulos de 30° , 60° , 120° e 150° e, caso a função meio estiver ligada, é feita a soma entre o raio do dispositivo semicircular (10 cm) e o valor lido por cada um dos pares de sensores $s1$ e $s7$.

No estado *mults* são registrados os valores calculados no estado anterior e, caso a função meio esteja ativada, o próximo estado será *smeio_1*, caso contrário a máquina vai para o estado *nmeio_1*.

Nos estados *smeio_1*, *smeio_2*, *smeio_3*, *nmeio_1*, *nmeio_y_1*, *nmeio_y_2*, *nmeio_x_1*, *nmeio_x_2* e *nmeio_pfim* são realizadas as operações restantes das equações Eq. (8) e Eq. (9) para calcular o vetor resultante, conforme a função escolhida previamente, ou seja, se a função meio está ou não ligada.

Finalmente, nos estados *nmeio_fim* e *smeio_fim* o sinal *done* é ativado no nível logo alto, indicando que a operação está concluída.

3.4.2.2 Comparador

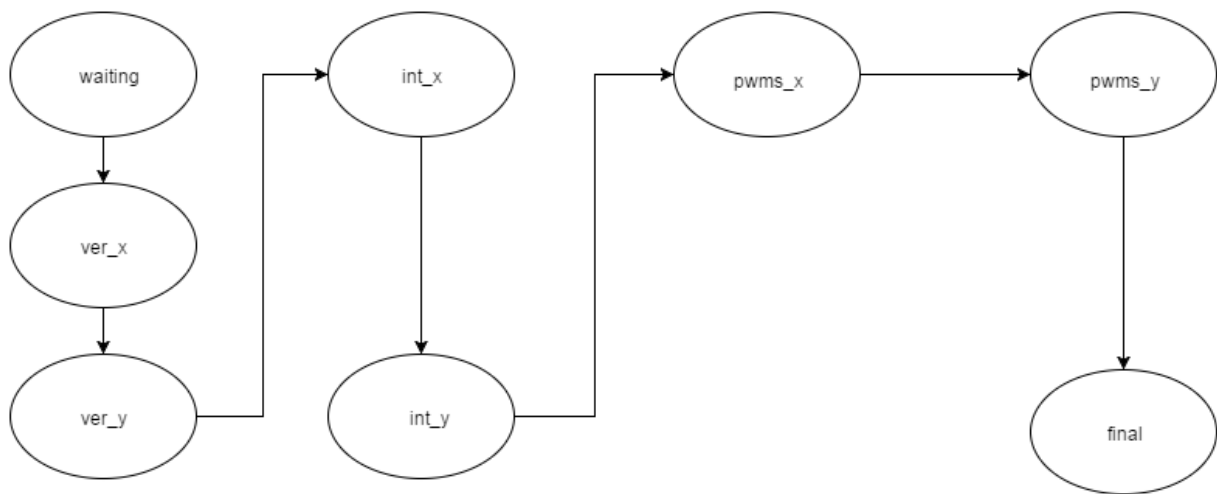


Figura 24. Diagrama de estados do comparador.

Conforme visto na seção 3.1, foram considerados 3 níveis de vibração para os atuadores piezoelétricos, que são: fraco, médio e forte. Comparando os vetores de reação provenientes do bloco anterior a faixa de valores que cada nível assume, este bloco produz como saídas sinais para ligar os atuadores piezoelétricos, assim como define um valor para alterar o *duty cycle* da onda PWM em cada nível.

No estado inicial *waiting*, são inicializados com o valor de “0” todos os sinais utilizados na arquitetura, que aguarda o sinal de inicialização do bloco.

Nos estados *ver_x* e *ver_y*, é verificado se os vetores resultantes nas direções x e y, respectivamente, ultrapassaram ou não a zona de proteção e, além disso, averigua para qual direção se deve acionar os atuadores piezoelétricos.

Nos estados *int_x* e *int_y* é realizada a verificação do nível de intensidade de vibração.

Nos estados *pwms_x* e *pwms_y* é feito o acionamento dos módulos PWM.

Finalmente, no estado final o sinal *done* é ativado no nível logo alto, indicando que a operação está concluída.

3.4.2.3 Arquitetura final

Finalmente, a Fig. (25) apresenta a proposta de solução para o sistema de *hardware* embarcado do dispositivo para deficientes visuais. A proposta de fusão sensorial é replicada para cada um dos 7 pares de sensores que enviarão simultaneamente os dados para o algoritmo de reação, que será responsável pelo processamento de informações, identificando um obstáculo perto do usuário do sistema.

Por fim, de acordo com essa última informação, o bloco deverá sensibilizar os dois atuadores piezoelétricos por meio de modulação *Pulse-Width Modulation* (PWM). Desta forma, esse sinal gerado deverá fazer com que o atuador piezoelétrico do lado contrário ao obstáculo vibre com mais intensidade à medida que o usuário se aproxima de uma colisão.

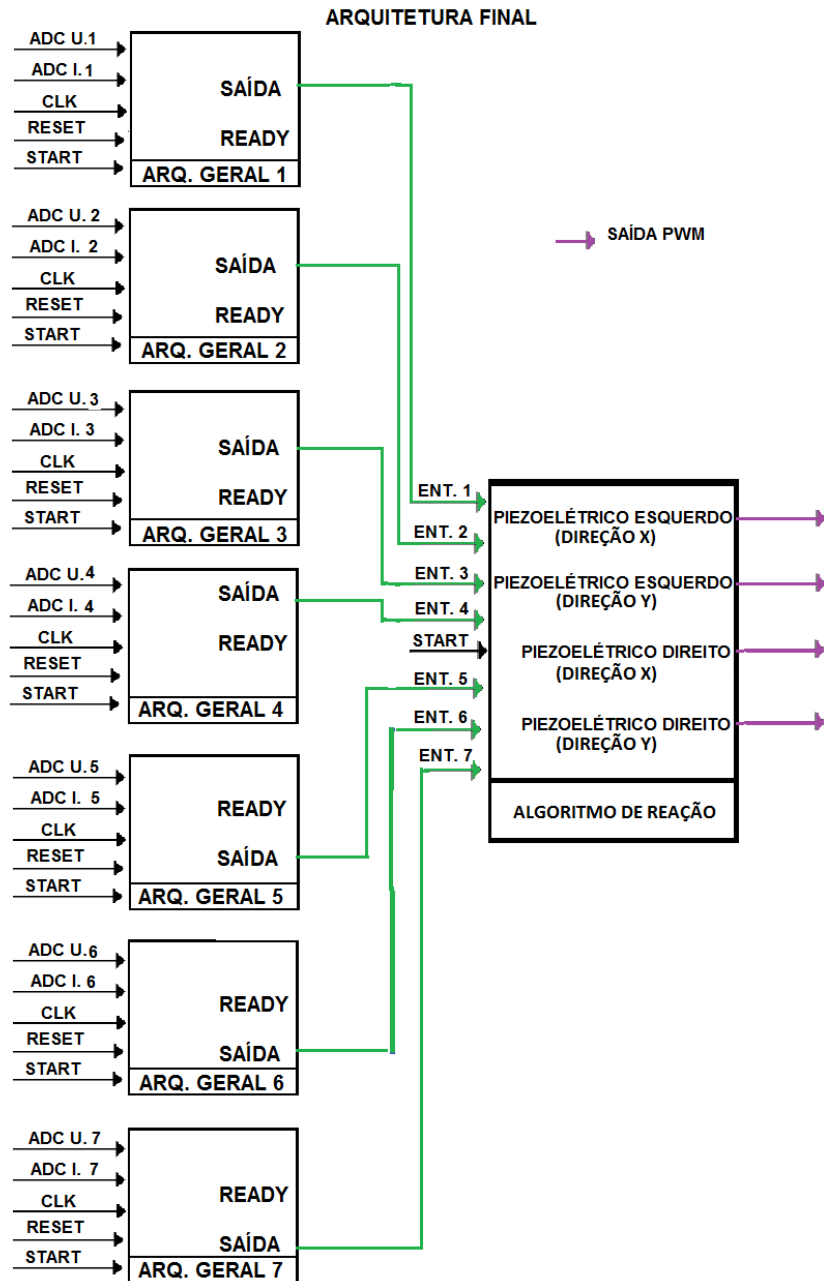


Figura 25. Arquitetura Final em diagrama de blocos.

4 METODOLOGIA E IMPLEMENTAÇÕES PARA TESTES NA FPGA

4.1 IMPLEMENTAÇÕES PARA TESTES NA FPGA

Para facilitar os testes na placa, propôs-se uma arquitetura simples para que fossem lidos os valores dos 7 sensores de ultrassom e os 7 valores dos sensores infravermelho nos *switches* da placa Nexys 4. A arquitetura tem o propósito de ler um número de 4 bits que representa um dos quatorze sensores, e as demais 10 chaves gravam os valores emulados dos sensores, que atribuirá o valor a um registrador quando pressionado um *push button*, conforme a Tab. (6) abaixo.

Após a gravação dos valores desejados, a arquitetura envia os valores aos blocos de fusão sensorial. Caso não seja atribuído um valor ao sensor pelas chaves, o código automaticamente atribui o valor correspondente a 150 cm, que significa que o sensor não está lendo nenhum obstáculo.

Tabela 6. – Codificação para cada sensor: U representa o sensor ultrassom e I o sensor infravermelho.

Código	Sensor
0001	U1
0010	I1
0011	U2
0100	I2
0101	U3
0110	I3
0111	U4
1000	I4
1001	U5
1010	I5
1011	U6
1100	I6
1101	U7
1110	I7

4.2 METODOLOGIA DE TESTES

Na etapa do TCC 1, os testes constituíram-se de comparações entre resultados gerados pelos modelos descritos em MATLAB e por simulação de *hardware*.

Na etapa do TCC 2, utilizou-se a mesma metodologia mencionada no parágrafo anterior para validação do bloco do algoritmo de reação.

Os resultados destas comparações são apresentados na seção posterior.

Para os testes no comparador, já integrado ao módulo PWM, foram utilizados *testbenches*, para simulação comportamental, e também foi usado o módulo descrito na seção anterior para testes na placa Nexys 4.

Os testes da arquitetura geral foram realizados diretamente na placa, sem simulações no software Xilinx ISE. Estipulou-se valores para os sensores simulando a presença de obstáculos dentro de uma zona de proteção de 50 cm.

É importante ressaltar aqui que nos módulos da filtragem de Kalman, foi necessário alterar o valor inicial da estimativa de distância \hat{x}_0 para 50 cm, a fim de convergir o resultado estimado para o resultado esperado mais rapidamente.

5 RESULTADOS

5.1 RESULTADOS DE SÍNTESE PARA IMPLEMENTAÇÕES EM VHDL

Nesta seção são apresentados os resultados de síntese para as implementações em VHDL de cada bloco da arquitetura completa para a fusão sensorial, considerando o consumo de registradores de *Flip-Flops (FFs)*, *Look-Up Tables (LUTs)* e blocos DSPs. São apresentados também resultados de desempenho dos circuitos propostos, como a frequência máxima de operação.

A Tabela (7) apresenta os resultados de síntese para as arquiteturas reconfiguráveis desenvolvidas neste projeto.

Tabela 7. – Resultados de síntese e performance das arquiteturas de *hardware*.

Arquitetura	Flip-Flops (Total de 126800)	LUTs (Total de 63400)	DSPs (Total de 240)	Freq. de Operação (MHz)
Arquitetura Geral da Fusão Sensorial	2036	4659	10	152.971
Modelo Ultrassom	114	401	1	273.269
Modelo Infravermelho	304	789	2	196.897
Filtro de Kalman	468	1018	2	152.971
Fusão Sensorial	495	1335	3	152.971
Algoritmo de reação	671	1944	1	196.897
Arquitetura Geral	14837	34943	78	152.971

5.2 COMPARAÇÃO ENTRE ARQUITETURAS DE *HARDWARE* E *SOFTWARE*

Nesta etapa, foram realizadas as comparações entre os resultados obtidos via *software* (MATLAB) e as arquiteturas de *hardware*. Para cada módulo da arquitetura, foi elaborado um gráfico entre o erro quadrático de cada amostra, dada pela Eq. (12). No fim, calculou-se o erro quadrático médio geral da arquitetura, que equivale à soma

de todos os erros quadráticos dividida pelo total de amostras, como aponta a Eq. (13). Os resultados são expostos nas seções seguintes.

$$EQ = (R_H - R_S)^2 \quad (12)$$

$$EQM = \frac{\sum_{i=1}^N EQ_i}{N} \quad (13)$$

Cada variável e seu significado está listada como se segue:

- a) EQ é o erro quadrático.
- b) EQM é o erro quadrático médio.
- c) R_H é o resultado de *hardware*.
- d) R_S é o resultado de *software*.
- e) EQ_i é a amostra i-ésima.
- f) N é a quantidade total de amostras.

A Tabela (8) mostra os valores de erro quadrático médio encontrados para as arquiteturas testadas em MATLAB e comparadas com *testbenches* gerados nas implementações em VHDL. Para os módulos dos modelos de sensor ultrassom e infravermelho, utilizou-se 90 amostras, varrendo distâncias entre 20 a 130 cm. Para os módulos do filtro de Kalman para os dois sensores e fusão sensorial, foram utilizadas 100 amostras. Para o modelo do algoritmo de reação, foram consideradas 60 amostras com um raio de proteção de 60 cm.

Tabela 8. – Erro quadrático médio das arquiteturas propostas.

Arquitetura	Erro quadrático médio
Arquitetura Geral da Fusão Sensorial	1,02
Modelo Ultrassom	2,53e-8
Modelo Infravermelho	3,39e-07
Filtro de Kalman para o sensor ultrassom	6,97e-04
Filtro de Kalman para o sensor infravermelho	1,71e-02
Fusão Sensorial	4,12e-08
Algoritmo de reação	1.29e-07

A seguir são apresentados os gráficos dos erros quadráticos para cada arquitetura.

5.2.1 Modelos do sensor ultrassom

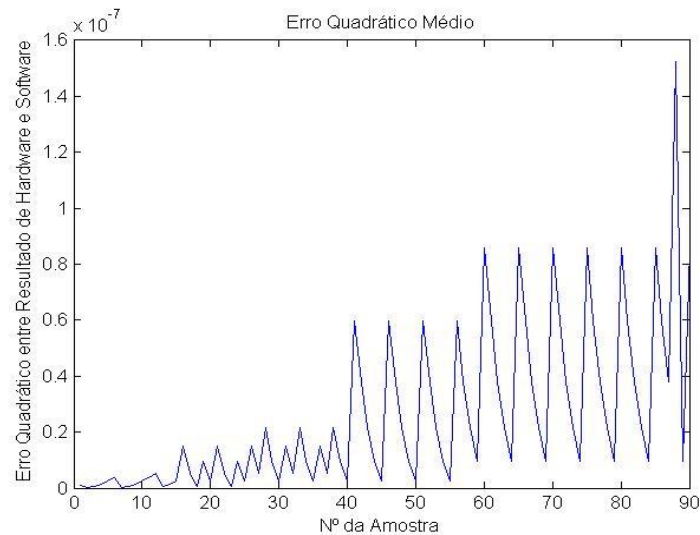


Figura 26. Erro quadrático médio entre resultados de *hardware* e *software* para os modelos do sensor ultrassom. $MSE = 2,53E-8$.

5.2.2 Modelos do sensor infravermelho

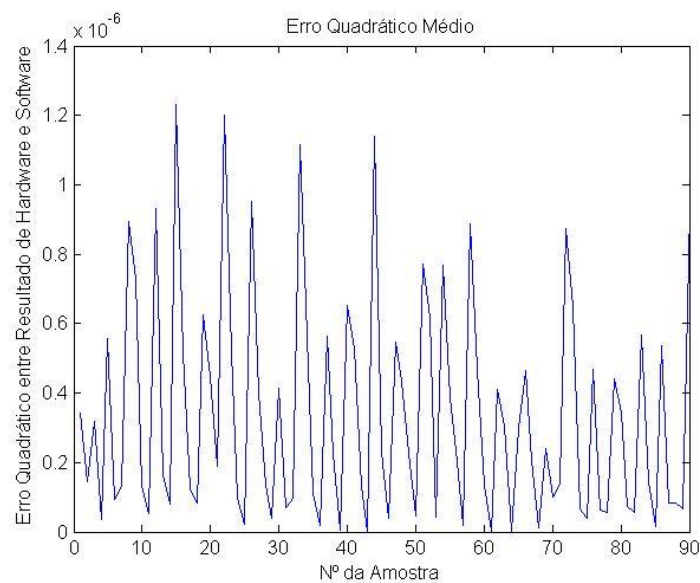


Figura 27. Erro quadrático médio entre resultados de *hardware* e *software* para o modelo do sensor infravermelho. $MSE = 3,39E-7$.

5.2.3 Modelos do algoritmo de FK

5.2.3.1 Algoritmo do FK integrado ao sensor ultrassom

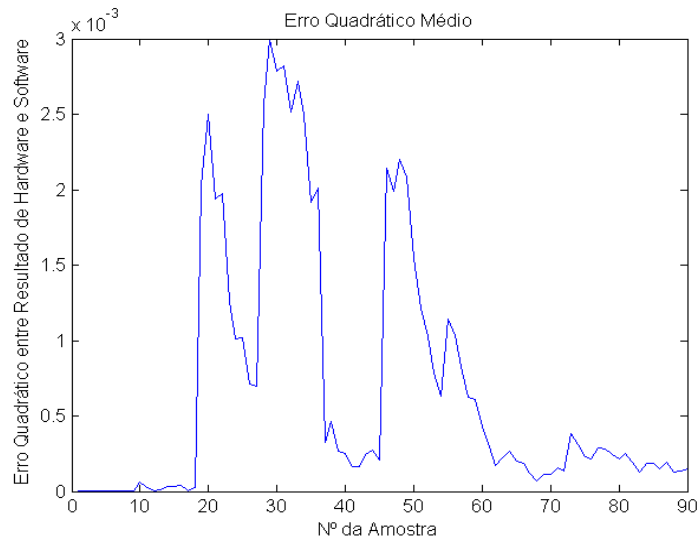


Figura 28. Erro quadrático médio entre resultados de *hardware* e *software* para o modelo do FK para o sensor ultrassom. MSE = 6,97E-4.

5.2.3.2 Algoritmo do FK integrado ao sensor infravermelho

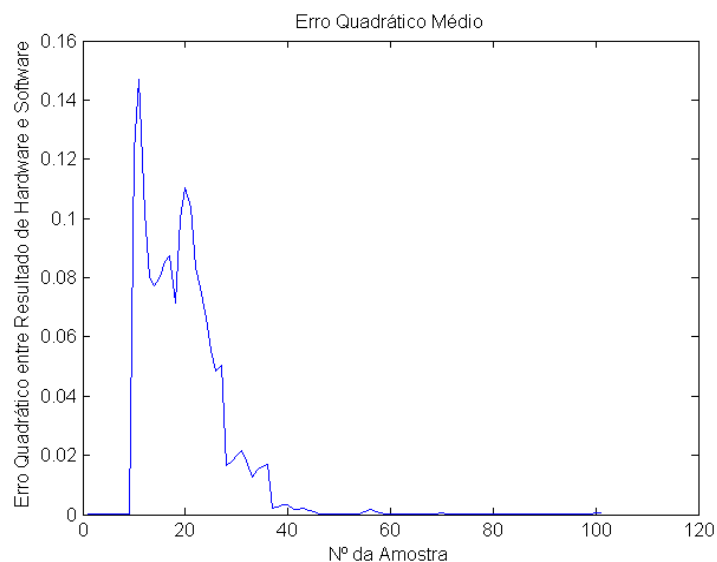


Figura 29. Erro quadrático médio entre resultados de *hardware* e *software* para o modelo do FK para o sensor infravermelho. MSE = 1,71E-2.

5.2.3.3 Análise e considerações sobre o módulo do filtro de Kalman

Foi observado, considerando os gráficos mostrados na Fig. (26) e Fig. (27), que esse módulo do filtro de Kalman não se mostrou tão preciso quanto os modelos para os sensores mostrados nas seções anteriores.

Isso pode ser explicado pela resolução da representação em ponto flutuante dos blocos de operações matemáticas e também pelo fato do algoritmo ser mais complexo que os demais.

Por tratar-se de um algoritmo recursivo, os parâmetros que influenciam diretamente o resultado da estimativa de medida são recalculados a cada instante e usados na iteração seguinte, como o erro de covariância e a própria estimativa. Isso causa acúmulo e propagação de erros, causando assim a diferença entre os resultados obtidos via *software* e *hardware*.

Entretanto, a nível de projeto, onde não é necessária tanta precisão, pode-se dizer que os resultados ainda são satisfatórios pelo fato dos erros estarem na ordem de fração de centímetros.

5.2.4 Modelos do algoritmo de fusão sensorial

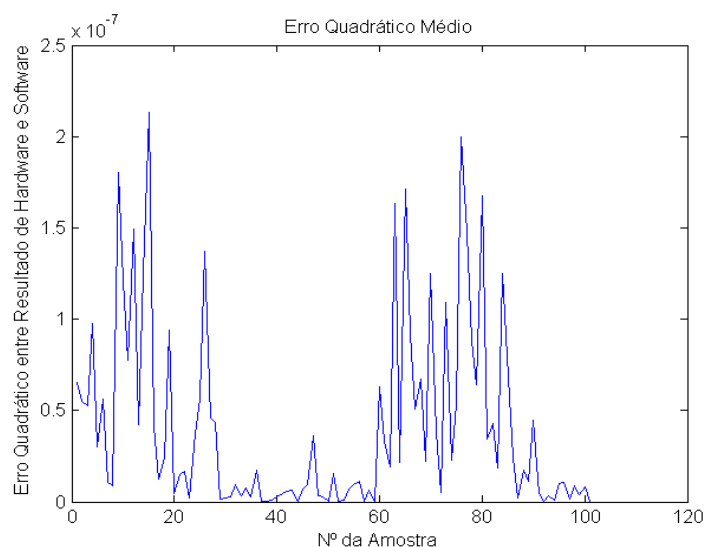


Figura 30. Erro quadrático médio entre resultados de *hardware* e *software* para o modelo de fusão sensorial. MSE = 4.12E-8.

5.2.5 Arquitetura geral da fusão sensorial

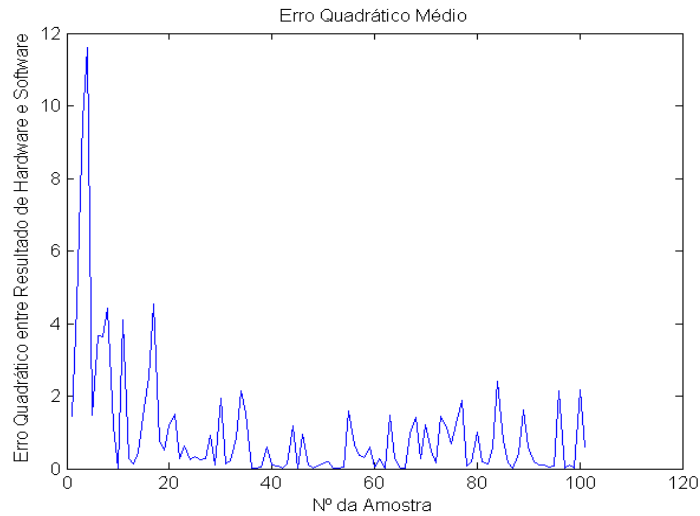


Figura 31. Erro quadrático médio entre resultados de *hardware* e *software* para a arquitetura geral de fusão sensorial. MSE = 1,02.

Na junção de todos os módulos, foram observados erros quadráticos bastante altos nas 5 primeiras amostras. Entre as amostras 5 a 20 o erro quadrático apresenta-se no limite da faixa tolerável de imprecisão e nos demais erros entre 0 a 2 cm.

A resolução do ponto flutuante, a complexidade de algoritmos e da arquitetura proposta e a propagação de erros podem explicar as diferenças significativas entre os resultados.

Os vetores de teste simularam os valores de saída dos conversores ADC para os dois sensores, ou seja, foram realizadas aproximações inteiras de valores fracionários, o que por si só afeta os resultados entre as simulações de *hardware* e *software*.

Conforme visto nas Fig. (26) e Fig. (27), a entrada dos módulos dos algoritmos do Filtro de Kalman já carregam um erro na ordem de $e-07$ cm, que são propagados e acumulados ao erro intrínseco do bloco para filtragem de ruído, conforme mostram as Fig. (28) e Fig. (29). Consequentemente, o bloco de fusão carrega nas suas entradas todos os erros propagados dos demais módulos da arquitetura geral, causando a discrepância entre os resultados de *hardware* e *software*.

5.2.5 Arquitetura do algoritmo de reação

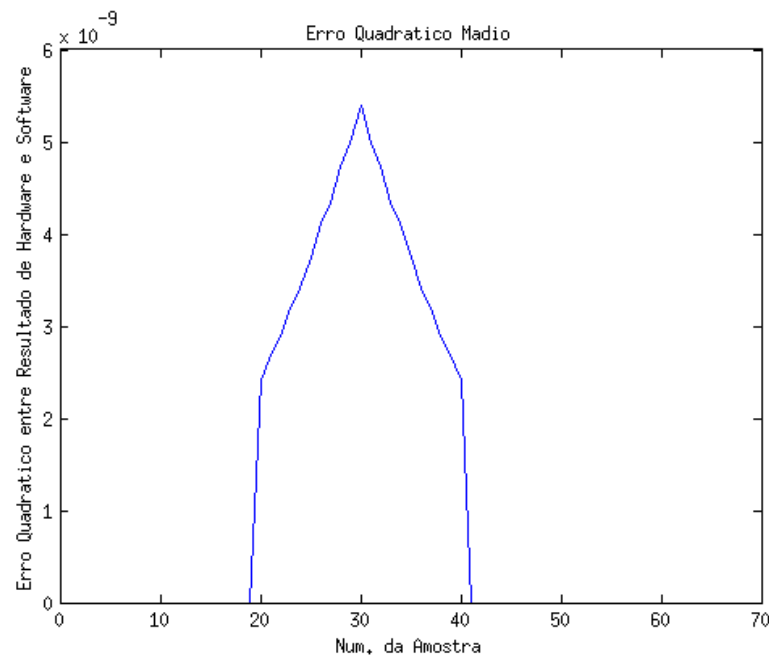


Figura 32. Erro quadrático médio entre resultados de *hardware* e *software* para a arquitetura do algoritmo de reação. MSE = 1.29E-7 .

5.3 TEMPO DE EXECUÇÃO

Nesta seção são apresentados na Tab. (9) o tempo de execução das implementações em arquiteturas de *hardware*, o qual se refere ao tempo em que o bloco recebe uma entrada e produz a saída final. Dado que, para a obtenção destes valores, é necessário a verificação do tempo via *testbench*, não foi possível obter o tempo de execução da arquitetura geral.

Tabela 9. – Tempo de execução das arquiteturas de *hardware*.

Arquitetura	Tempo de Execução (ns)
Modelo Ultrassom	70
Modelo Infravermelho	130
Filtro de Kalman	210
Fusão Sensorial	220
Algoritmo de reação	215

6 CONCLUSÕES

Este trabalho apresentou propostas de arquiteturas de *hardware* para um sistema eletrônico embarcado com algoritmos de tratamento de dados para estimação de distâncias entre usuário e obstáculos, que futuramente poderá ser embutido em um dispositivo com formato semicircular que visa auxiliar a locomoção de deficientes visuais, indicando a direção de movimento a fim de evitar colisões contra paredes, objetos ou pessoas.

Foram propostos como algoritmos de tratamento de dados o filtro de Kalman, aplicado à filtragem de ruídos e fusão sensorial. Conforme apresentado nos capítulos 1 e 2, esses processos se mostram como uma das soluções mais aplicadas e recomendadas ao contexto de filtragem de sensores.

Nas implementações dos algoritmos FK e fusão sensorial em MATLAB, verificou-se o comportamento esperado na teoria. Nas Figuras (14) e (15) observou-se que de fato o FK comporta-se como um bom filtro de ruídos, pois à medida que se aumentava o número de amostras, cada vez mais a variável estimada se aproximava do valor real de medida. Na Figura (16) também foi observado o comportamento esperado pela teoria, já que a fusão sensorial priorizou a medida de um sensor e o resultado final sempre convergia para valores mais próximos ao sensor ultrassom.

Nas arquiteturas de *hardware* implementadas, esperava-se valores próximos àqueles encontrados em simulações de *software*, com pequenos erros associados. O erro quadrático médio mostra-se como um bom estimador do quão distante estão esses valores. Observou-se, para os modelos dos sensores ultrassom e infravermelho e para fusão sensorial, um erro bastante pequeno, validando a proposta de solução apresentada.

Na arquitetura do algoritmo do filtro de Kalman, os erros quadráticos médio mostraram-se maiores que nos demais módulos. Ainda assim, por estarem na ordem de fração de centímetros, mostraram-se adequados para a implementação proposta por este projeto, já que o mesmo não requisita grande precisão.

Entretanto, na união de todos os módulos para a implementação da arquitetura geral de fusão sensorial, o erro quadrático médio apresentou-se alto em comparação

com todos os módulos, já na ordem de cm. Apesar de ser considerado tolerável tendo em vista o requisito de projeto, em termos de simulação o comportamento apresentado é indesejado.

Foi proposto um algoritmo simples para tratamento dos dados provenientes da etapa de fusão sensorial, que é responsável por reagir à entrada de um obstáculo em uma zona imaginária de proteção. Esse algoritmo deve ser capaz de acionar vibradores piezoelétricos com intensidade proporcional à distância que o usuário deve percorrer para retirar o obstáculo da zona de proteção.

Foram realizadas simulações no MATLAB para validação de proposta, e constatou-se que o algoritmo funcionava de acordo com o esperado. Após a codificação em VHDL, foram feitos testes na placa Nexys 4, simulando os valores lidos pelos 7 pares de sensores através das entradas nos *switches* da placa. Observou-se o comportamento esperado da arquitetura durante esses testes, pois os LEDs da placa alteraram seu brilho de maneira correta para diferentes níveis de entrada.

Conclui-se por meio das implementações realizadas que o dispositivo é um detector razoável de obstáculos, principalmente quando o raio de proteção é pequeno, pois diminui o espaço entre dois pares de sensores, podendo captar com maior precisão os obstáculos, sendo assim, afirma-se que todos os objetivos propostos neste trabalho foram alcançados com êxito.

Notou-se, entretanto, que o algoritmo e o dispositivo possuem suas limitações, que são a não detecção de obstáculos entre dois pares de sensores, o que pode levar a choque com obstáculos e as suposições que o usuário somente realize sua caminhada para frente, o que não condiz com a realidade, na qual um usuário pode andar na diagonal. Além disso, o dispositivo só é capaz de detectar obstáculos na altura da cabeça do usuário, o que pode ser um problema já que podem existir obstáculos pequenos ou abaixo da altura da cabeça do usuário, que causam colisões.

Portanto, para trabalhos futuros, propõe-se uma melhoria nos algoritmos de tratamentos de dados de reação à obstáculos com algoritmos de maior grau de complexidade, capaz de detectar um maior número de obstáculos, e um aumento do número de sensores utilizados no dispositivo, ou pela posição que é usado no corpo do deficiente físico.

REFERÊNCIAS

- Aiube, F.A.L., 2005, “Modelagem dos preços futuros de Commodities: abordagem pelo filtro de partículas”, Tese (Programa de Pós-Graduação em Engenharia de Produção) - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Disponível em: <http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0124838_05_cap_04.pdf>.
- Brentano, L., 2011, “Brasileiro cria bengala eletrônica de baixo custo para deficientes visuais”, G1, São Paulo, Tecnologia da Universidade de Brasília, Brasília, Disponível em: <<http://g1.globo.com/tecnologia/noticia/2011/07/brasileiro-cria-bengala-eletronica-de-baixo-custo-para-deficientes-visuais.html>>.
- Conde, M.E. et al., 2013, “An efficient data fusion architecture for infrared and ultrasonic sensors using FPGAs”, IEEE Proc. of the Latin American Symposium on Circuits and Systems, Cusco, Peru, p. 1-4.
- Digilent, Nexys 4 image, Disponível em: <<http://www.digilentinc.com/Data/Products/NEXYS4/Nexys4-obl-600.png>>.
- Diniz, E.H., Demarchi, F.M., 2013, “Desenvolvimento de um sistema para auxílio à locomoção de deficientes visuais através da implementação em arquiteturas reconfiguráveis da transformada Census para estimação de distância usando visão estéreo”, Monografia (Graduação em Engenharia Eletrônica) - Universidade de Brasília, Brasília, 156 p.
- Gharpure, C., Nicholson, J., Pavithran, S., 2004, “RFID in robot-assisted indoor navigation for the visually impaired”. IEEE Proc. of the International Conference on Intelligent Robots and Systems, Sendai, Japan, p. 1979-1984.
- GP2Y0A02YK, 2nd ed., Sharp, September 2001.
- Hersh, M.A., Johnson, M.A., 2008, “Assistive technology for visually impaired and blind people”, Springer-Verlag.
- Hoefer, S., 2011, “Meet the tacit project. It's sonar for the blind”, Disponível em: <<http://grathio.com/2011/08/meet-the-tacit-project-its-sonar-for-the-blind/>>.
- Hoffmann, S.B., Seewald, R., 2003, “Caminhar sem medo e sem mito: orientação e mobilidade”, Disponível em: <<http://www.bengalalegal.com/orienta>>.
- Kalman, R.E., 1960, “A New Approach to Linear Filtering and Prediction Problems”, Transactions of the ASME – Journal of Basic Engineering, n. 82 (Series D), p. 35-45, Disponível em: <<http://www.cs.unc.edu/~welch/kalman/media/pdf/Kalman1960.pdf>>.

- Kilts, S., 2007, "Advanced FPGA Design: Architecture, Implementation, and Optimization", Minneapolis, Minnesota.
- Kim, P., 2011, "Kalman filter for beginners: with matlab examples", Trad.: Lynn Huh. CreateSpace Independent Publishing Platform.
- Maxsonar LV-EZ1-MB1010, 1st ed., MaxBotix Inc., 2005.
- Meyer-Baese, U., 2004, "Digital signal processing with field programmable gate arrays", Springer-Verlag.
- Morris, A.S., 2001, "Measurement and instrumentation principles", BH, London.
- Muñoz, D.M. et al., 2010, "Robô cão guia para auxílio a deficientes visuais em recintos fechados", VI Congresso Nacional de Engenharia Mecânica (VI National Congress of Mechanical Engineering), Campina Grande, Paraíba – Brasil, Disponível em: <<http://www.abcm.org.br/pt/wp-content/anais/conem/2010/PDF/CON10-1860.pdf>>.
- Naiff, D. et al., 2011, "Medição de distância por infravermelho", Instituto de Estudos Superiores da Amazônia (IESAM), Disponível em: <<http://www3.iesampa.edu.br/ojs/index.php/computacao/article/viewFile/843/617>>.
- National Instruments, 2013, "Fundamentos da tecnologia FPGA", Disponível em: <<http://www.ni.com/white-paper/6983/pt/>>.
- Negenborn, R., 2003, "Robot Localization and Kalman Filters: On finding your position in a noisy world. Thesis (Master of Science, specialized in Intelligent Systems) - Institute of Information and Computing Sciences, Copenhagen, Disponível em: <http://www.negenborn.net/kal_loc/thesis.pdf>.
- Oliveira, T.A., Morais, A.R., Cirillo, M.A., 2011, "Métodos de estimação de parâmetros em modelo de covariância com erro na covariável", Ciência Rural, Santa Maria, v. 41, n. 10, p. 1851-1857, Disponível em: <<http://www.scielo.br/pdf/cr/v41n10/a14311cr3850.pdf>>.
- Organização Mundial da Saúde, 2013, "A deficiência visual e cegueira", Fact Sheet n. 282, Disponível em: <<http://www.who.int/mediacentre/factsheets/fs282/en/>>.
- Paim, P.K., 2005, "Modelamento, simulação e controle de veículos autônomos aéreos e submarinos", Monografia (Graduação em Engenharia Mecatrônica) - Faculdade de Tecnologia da Universidade de Brasília, Brasília, Disponível em: <<http://lara.unb.br/~gaborges/arquivos/pf.pedro.paim.2005.1.pdf>>.
- Portal Terra, 2011, "Japoneses criam novo robô cão-guia para deficientes visuais", Tecnologia, Disponível em: <<http://tecnologia.terra.com.br/robos/japoneses->

criam-novo-robo-caio-guia-para-deficientes-visuais,7499e194c2bda310VgnCLD200000bbcceb0aRCRD.html>.

Robotics-electronics, Infrared sensor image. Disponível em:

<<http://www.robot-electronics.co.uk/media/catalog/product/cache/1/image/650x/040ec09b1e35df139433887a97daa66f/2/y/2y0a02.jpg>>.

Roentgen, U.R. et al., 2008, "Inventory of electronic mobility aids for persons with visual impairments: a literature review", Journal of Visual Impairment & Blindness, Vol. 2012, No. 11, p. 702-724.

Santos, C.H.F., [s.d.], "Teoria e prática do filtro de Kalman", Aula 1, Disponível em: <<http://www.foz.unioeste.br/~chsantos/Extensao/FK/AULA01.pdf>>.

Santos, D.R.G. et al., 2010, "Desenvolvimento de uma bengala eletrônica para locomoção de pessoas com deficiência visual", VI Congresso Nacional de Engenharia Mecânica (VI National Congress of Mechanical Engineering), Campina Grande, Paraíba, Brasil, Disponível em: <<http://www.abcm.org.br/pt/wp-content/anais/conem/2010/PDF/CON10-0608.pdf>>.

Sass, R., Schmidt, A., 2010, "Embedded systems design with platform FPGA, principles and practices", Morgan Kaufmann.

Singhal, H., Kaur, A., Yadav, R., 2013, "The development of an intelligent aid for blind and old people", Emerging Trends and Applications in Computer Science (ICETACS), 1st International Conference, p. 182, 186.

Sullivan, P., 2013, "Com custo de até R\$ 138 mil, treinar cães-guia é desafio para ONGs nos EUA", Disponível em: <<http://ultimosegundo.ig.com.br/mundo/nyt/2013-11-10/com-custo-de-ate-r-138-mil-treinar-caes-guia-e-desafio-para-ongs-nos-eua.html>>.

Vahid, F., 2007, "Digital design", United States of America.

ANEXO

Anexo I – pwm.vhd

```

-----
-----
--
--   FileName:          pwm.vhd
--   Dependencies:      none
--   Design Software:    Quartus II 64-bit Version 12.1 Build 177 SJ Full
Version
--
--   HDL CODE IS PROVIDED "AS IS." DIGI-KEY EXPRESSLY DISCLAIMS ANY
--   WARRANTY OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT
--   LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
--   PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL DIGI-KEY
--   BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL
--   DAMAGES, LOST PROFITS OR LOST DATA, HARM TO YOUR EQUIPMENT, COST OF
--   PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY OR SERVICES, ANY CLAIMS
--   BY THIRD PARTIES (INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF),
--   ANY CLAIMS FOR INDEMNITY OR CONTRIBUTION, OR OTHER SIMILAR COSTS.
--
--   Version History
--   Version 1.0 8/1/2013 Scott Larson
--       Initial Public Release
--   Version 2.0 1/9/2015 Scott Larson
--       Transistion between duty cycles always starts at center of pulse to
avoid
--       anomalies in pulse shapes
--
-----
-----

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY pwm IS
  GENERIC(
    sys_clk          : INTEGER := 50_000_000; --system clock frequency in
Hz
    pwm_freq         : INTEGER := 100_000;    --PWM switching frequency in
Hz
    bits_resolution  : INTEGER := 8;          --bits of resolution setting
the duty cycle
    phases           : INTEGER := 1);         --number of output pwms and
phases
  PORT(
    clk              : IN  STD_LOGIC;          --
system clock
    reset_n          : IN  STD_LOGIC;          --
asynchronous reset
    ena              : IN  STD_LOGIC;          --
latches in new duty cycle
    duty             : IN  STD_LOGIC_VECTOR(bits_resolution-1 DOWNTO 0); --duty
cycle
    pwm_out          : OUT STD_LOGIC_VECTOR(phases-1 DOWNTO 0);    --pwm

```

```

outputs
    pwm_n_out : OUT STD_LOGIC_VECTOR(phases-1 DOWNT0 0));      --pwm
inverse outputs
END pwm;

ARCHITECTURE logic OF pwm IS
    CONSTANT period      : INTEGER := sys_clk/pwm_freq;
--number of clocks in one pwm period
    TYPE counters IS ARRAY (0 TO phases-1) OF INTEGER RANGE 0 TO period - 1;
--data type for array of period counters
    SIGNAL count          : counters := (OTHERS => 0);
--array of period counters
    SIGNAL half_duty_new   : INTEGER RANGE 0 TO period/2 := 0;
--number of clocks in 1/2 duty cycle
    TYPE half_duties IS ARRAY (0 TO phases-1) OF INTEGER RANGE 0 TO period/2;
--data type for array of half duty values
    SIGNAL half_duty       : half_duties := (OTHERS => 0);
--array of half duty values (for each phase)
BEGIN
    PROCESS(clk, reset_n)
    BEGIN
        IF(reset_n = '0') THEN
--asynchronous reset
            count <= (OTHERS => 0);
--clear counter
            pwm_out <= (OTHERS => '0');
--clear pwm outputs
            pwm_n_out <= (OTHERS => '0');
--clear pwm inverse outputs
            ELSIF(clk'EVENT AND clk = '1') THEN
--rising system clock edge
                IF(ena = '1') THEN
--latch in new duty cycle
                    half_duty_new <= conv_integer(duty)*period/(2**bits_resolution)/2;
--determine clocks in 1/2 duty cycle
                END IF;
                FOR i IN 0 TO phases-1 LOOP
--create a counter for each phase
                    IF(count(0) = period - 1 - i*period/phases) THEN
--end of period reached
                        count(i) <= 0;
--reset counter
                        half_duty(i) <= half_duty_new;
--set most recent duty cycle value
                    ELSE
--end of period not reached
                        count(i) <= count(i) + 1;
--increment counter
                    END IF;
                END LOOP;
                FOR i IN 0 TO phases-1 LOOP
--control outputs for each phase
                    IF(count(i) = half_duty(i)) THEN
--phase's falling edge reached
                        pwm_out(i) <= '0';
--deassert the pwm output
                        pwm_n_out(i) <= '1';
--assert the pwm inverse output
                    ELSIF(count(i) = period - half_duty(i)) THEN
--phase's rising edge reached
                        pwm_out(i) <= '1';

```

```
--assert the pwm output
    pwm_n_out(i) <= '0';
--deassert the pwm inverse output
    END IF;
    END LOOP;
    END IF;
    END PROCESS;
END logic;
```